Intelligent X-ray Imaging Inspection System for the Food Industry

by
Catalin Gheorghe Amza

Supervisors:
Dr. Peter R. Innocent
Prof. Jeffrey A.G. Knight

Thesis Submitted in Partial Fulfilment
of the Requirements of De Montfort University
for the Degree of Doctor of Philosophy

Faculty of Computing Sciences and Engineering

Leicester
2002

ABSTRACT

The inspection process of a product is an important stage of a modern production factory. This research presents a generic X-ray imaging inspection system with application for the detection of foreign bodies in a meat product for the food industry. The most important modules in the system are the image processing module and the high-level detection system.

This research discusses the use of neural networks for image processing and fuzzy-logic for the detection of potential foreign bodies found in x-ray images of chicken breast meat after the de-boning process. The meat product is passed under a solid-state x-ray sensor that acquires a dual-band two-dimensional image of the meat (a low- and a high energy image). A series of image processing operations are applied to the acquired image (pre-processing, noise removal, contrast enhancement). The most important step of the image processing is the segmentation of the image into meaningful objects. The segmentation task is a difficult one due to the lack of clarity of the acquired X-ray images and the resulting segmented image represents not only correctly identified foreign bodies but also areas caused by overlapping muscle regions in the meat which appear very similar to foreign bodies in the resulting x-ray image. A Hopfield neural network architecture was proposed for the segmentation of a X-ray dual-band image. A number of image processing measurements were made on each object (geometrical and grey-level based statistical features) and these features were used as the input into a fuzzy logic based high-level detection system whose function was to differentiate between bones and non-bone segmented regions. The results show that system's performance is considerably improved over non-fuzzy or crisp methods. Possible noise affecting the system is also investigated.

The proposed system proved to be robust and flexible while achieving a high level of performance. Furthermore, it is possible to use the same approach when analysing images from other applications areas from the automotive industry to medicine.

to my mum
and dad

AKNOWLEDGEMENTS

I am deeply in debt to both my supervisors Peter Innocent and Jeffrey Knight who have given me their support, comments and constructive suggestions to improve my research. Thanks are due to Jeff Knight without who this research would not have been possible.

Special thanks are due to my mum and dad who supported and encouraged me along the preparation of this research. They taught me that self-motivation and never-ending desire for knowledge are important for living a successful life.

Last but no least, I wish to thank all my friends who provided consistent and constant moral support that kept me going throughout this research.

"If we knew what we were doing,
it wouldn't be called research,
would it?"

Albert Einstein
1879-1955

CONTENTS

GLOSSARY

TERM	DEFINITION
Antecedent	Defines the combined truth of state of the current fuzzy region for a particular variable
Background removal	Image processing function for removing the biased frame from an image.
Cluster	Group of objects with the same identification
Consequent	The action or right-hand side part of a rule.
Contrast	Measure of the distribution of grey levels in an image
Convolution	Linear mathematical process
Crisp set or value	Crisp sets have distinct and sharply defined membership edges (classical sets where membership is either 1 for totally contained in the set or 0 for totally excluded from the set).
Defuzzification	The process of deriving a scalar representing a control variable's expected value from a fuzzy set.
Degree of membership	This is the degree to which a variable's value is compatible with the fuzzy set.
Detection	Process intended to establish the presence of certain objects or foreign bodies in an image
False negatives	The proportion of positive cases that test negative
False positives	The proportion of negative cases that test positive
Feature	Quantity characterizing an image or its objects
Fuzziness	The degree or quality of imprecision intrinsic in a property, process or concept.
Fuzzy set	It differs from the crisp sets by allowing partial or gradual memberships.
Grey level	Measure of intensity of an image
Histogram	Graphical means for specification of frequencies (statistical).
Image enhancement	Image processing method altering the histogram of an image towards a broader level distribution

Linguistic variable	In the rule formulation language of fuzzy systems, the term applied to a fuzzy set or the combination of a fuzzy set and its associated hedges.
Noise	Irrelevant signal superimposed upon a given signal
Recognition	Process intended to identify a given image or object with a reference image or object
Rule	Statement of knowledge that relates the compatibility of fuzzy premise propositions to the compatibility of one or more consequent fuzzy spaces.
Segmentation	Partitioning an image into meaningful objects or segments
Sensitivity	The proportion of positive cases that test positive
Specificity	The proportion of negative cases that test negative
Test	Method for the accepting or rejecting of a hypothesis

ACRONYMS

2D = two dimensional
3D = three dimensional
AHC = agglomerative hierarchical clustering
ANN = artificial neural network
BGA = ball grid array
BP = back-propagation learning algorithm
BPNN = back-propagation-based neural network
CHNN = competitive Hopfield neural network
CMMNN = Competitive Multi Module Neural Network
CSNN = constraint satisfaction neural network
CSP = constraint satisfaction problem
CT = computer tomography image
FCM = Fuzzy C-Means clustering method
FFNN = feed forward neural network
FKNN = fuzzy K-nearest neighbour algorithm
FL = fuzzy logic
FN = false negative
FNF = false negative fraction
FP = false positive
FPF = false positive fraction
GA = genetic algorithm
HCM = hard C-means algorithm
HNN = Hopfield neural network
KNN = Kohonen neural network
LEGION = locally excitatory globally inhibitory oscillator network
MFCM = modified hard C-means algorithm
MLFF = multi-layer feed forward neural network
MLHNN = multi-layer Hopfield neural network
MLP = multi-layer perceptron
MRI = magnetic resonance image
MS-CSNN = multi-scan constraint satisfaction neural network
NN = neural network
PCB = programmable control board
PCNN = pulse-coupled neural network
PE = pulmonary emboli
PR = pattern recognition
PSOM = probabilistic self-organising map
ROC = receiver operator characteristic
SMLP = single multi layer perceptron
SOM = self-organising map (Kohonen)
TN = true negative
TNF = true negative fraction
TP = true positive
TPF = true positive fraction
VLSI = very large scale integrated circuit
WTA = winner-takes all learning mechanism

NOMENCLATURE

TERM	DEFINITION
R/C	Rayleigh-to-Compton ratio
I	Image
(x,y)	Pixel at location x and y
f(x,y)	Intensity value of the pixel
S	Segments
E	Energy for HNN
w_{ij}	Weight between neuron i and neuron j for HNN
v_i	Output of neuron i^{th} for HNN
I_i	External input for the i^{th} neuron for UNN
h_x	Histogram value for the x^{th} grey-level for HNN
d_{xy}	Euclidian distance between grey-level x and y
R_{pi}	Distance of point I from centroid of class p
U	Membership matrix for FCM algorithm
$u_{x,i}$	Element in the above matrix U for FCM algorithm
z_i	Class centers for FCM algorithm
g(x,y)	Processed image
h(x,y)	Position-invariant operator
F(u,v)	Fourier tranform of f(x,y)
(u,v)	Frequecy-domain correspondents for spatial domain values (x,y)
G(u,v)	Fourier transform of g(x,y)
$G_{f(x,y)}$	Gradient of an image
$L_{f(x,y)}$	Laplacian of an image
net_i	Total input to the i^{th} neuron for the HNN
t	Time

N	Number of neurons in the HNN architecture
n by n	Number of pixels in the image (HNN aproach)
k	Number of classes for the HNN architecture (HNN aproach)
l_x, l_y	Grey-level values (HNN aproach)
DIS_{xy}	Euclidian distance between grey-level l_x and l_y (HNN aproach)
N_1, N_2	The number of grey-levels present in the low- and high-energy images (HNN aproach)
N	Maximum number of levels from the low- and high-energy images (HNN aproach)
hh_x, hl_y	Histogram values for low- and high-energy images for grey level x and y respectively (HNN aproach)
$V_{x,i}$	Output of neuron (x,i) after WTA algorithm was applied (HNN aproach)
$f_n(x,y)$	Image corrupted with noise
n(x,y)	Additive random noise
$E^{(t+1)}$, $E^{(t)}$	Energy associated with the HNN network at time (t+1) and (t)
Err	Mean square error for the BP algorithm
w1[]	Array of weights from input to hidden layer (BPNN)
w2[]	Array of weights from hidden to output layer (BPNN)
x[]	Array of input values (BPNN)
h[]	Array of hidden nodes (BPNN)
y	Output of the BPNN network
η	Learning rate for the BP learning algorithm
Output_error, hidden_error	The error at the output and hidden layer respectively
A	Fuzzy set (FL)
$\mu_A(x)$	Membership function of x in A (FL)
X	Universe of discourse (FL)
x	Fuzzy variable
Y	Linguistic variable

LIST OF FIGURES

LIST OF TABLES

"Everything should be as simple as it is, but no simpler."

Albert Einstein
1879-1955

C H A P T E R 1

INTRODUCTION

1.1 Background

In the production of meat products a most common complaint is the presence of bones in the final, supposedly bone-free, product. Although this is a problem for all meat production it is more of a problem in poultry production compared with beef, pork and lamb. The reason for this is that the chicken carcass is much softer than that of cattle and therefore it is more likely that a bone will break off during the de-boning process. The presence of bones is considered a far greater problem in products such as chicken sandwiches and ready meal products where, because the consumer eats the product with little or no further preparation, the perception that the product should be bone free is greater. Although the occurrence of bones in a supermarket tray packed with chicken breast portions is a concern, it does not as yet attract the same concern as bones in ready meals or fast food take away meals. The occurrence of bones is far greater (up to ten times for impacted wish bones) in those processing plants, which have automated chicken de-boning equipment rather than the traditional but labour intensive method of hand de-boning. As more processing plants become automated, the occurrence of bones is set to increase. Until recently the only inspection technology available was that of manually inspecting every chicken fillet for bone. In almost every

chicken processing plant in the world dozens of people stand at the end of the production line literally feeling every chicken piece by hand for the presence of bone. Such a monotonous task coupled with the fact that after only a few minutes the inspector's hands become extremely cold means that many bones, which should be detected, are allowed to pass through. In addition there are many bones, which are fully impacted in the meat, which are impossible to detect by hand even for a well-trained and observant inspector. It is therefore not surprising that recently there has been a great deal of interest in the development of technology in order to automate the inspection task.

To date the most common system that has been used for the inspection of bones in meat products has been based on the use of X-rays. X-ray transmission systems rely on X-rays passing through the product under inspection and impinging on a sensor, which is responsive to X-rays. The system used for the experiments in this work was the proprietary IMS[1] BoneScan™ system. In this system, as the meat passes over the sensor, a complete two-dimensional real-time x-ray image is acquired. The image obtained from such a system may be complicated by the fact that parts of the meat can fold back on itself. This is typical in real poultry plants where the demands of high volume throughput mean that it is impossible to align pieces of meat perfectly uniformly which very often happens in laboratory based experimentations. Most commercial systems currently rely on simple thresholding of the image in order to identify foreign bodies. The result of thresholding is that the image is separated into different regions of brightness and some of these correspond to areas which are possible bones. This process is known as image segmentation. This is only effective in applications where the meat is of uniform and constant thickness such as the inspection of chicken nuggets after the nugget forming stage. Thus, in this case when used with raw chicken meat, segmentation (and consequently, bone detection) can be a very difficult process.

There have been reported attempts to use neural networks for bone detection but they suffered from poor performance and generalisation. Previous academic research work is presented in *[Graves, 1999]*. However, Grave's research is mainly focused on the design and implementation of the hardware for an X-ray imaging inspection system for the food industry. Image processing of the acquired images is done using

morphological filtering, and the high-level detection is performed using a simple look-up table *[Graves, 2000, personal communication]*.

The research reported in this work is a development of an investigation successfully undertaken and completed for the M.Sc. Human-Computer Systems ("Neural Network Classification of Bones from X-ray images of Chicken Meat", *[Amza, 1999]*).

The research methodology applied here was to first collect image data (using appropriate equipment donated to De Montfort University by the SFT[2]) and examine the variations in foreign object appearance based on previous project work completed on the M.Sc. Human-Computer Systems. According to the variations and their sources, various artificial intelligence and image processing techniques are investigated in order to produce an optimal inspection system that can be used to determine the presence or absence of foreign bodies.

1.2 Research objective and methodology

The main aim of this research is: to develop a generic intelligent X-ray imaging inspection system with applicability to the food industry. The research methodology methods are investigations of artificial methods (neural networks, fuzzy logic, genetic algorithms, etc.) for image processing of the acquired X-ray images are performed in order to propose a system with a minimum human intervention, a minimum number of parameters and with a high degree of performance accuracy, flexibility and robustness.

Classical and artificial intelligence image processing methods are investigated for image enhancement, segmentation and feature extraction so that further image analysis and foreign body detection processes can be facilitated. The main objective is to develop and evaluate a flexible software system for a generic inspection system.

[1] Intelligent Manufacturing Systems ltd. (IMS)
[2] Spectral Fusion Technologies (former Intelligent Manufacturing Systems ltd), Coleshill, Birmingham

1.3 Organisation of Thesis

The research opens with a small introduction presenting the background, reasons behind and the objectives of this research.

Chapter 2 starts with presenting a generic X-ray inspection system from the hardware point of view. Then, the X-ray inspection system used for the experiments in this research is presented, along with other existing methods for detection of foreign bodies in the context of food processing. The chapter concludes with a review of the current state of commercially available systems for food processing.

Based on the conclusions of the previous chapter about the difficulty of segmentation of an X-ray image taken from a meat product, a literature review of available methods is presented in Chapter 3. The classification of segmentation methods is summarised as classical and artificial intelligence methods. The emphasis in this chapter is on critically reviewing artificial intelligence methods since this is a focus of this research. Neural networks, genetic algorithms and fuzzy logic approaches to segmentation are taken into consideration and discussed. Finally, conclusions are drawn and recommendations for further image analysis are made accordingly.

The developed X-ray imaging system is discussed in Chapter 4. The image analysis methods presented here include low level (pixel) processing, image segmentation, and high-level (object) detection. The acquired X-ray image (a dual-band consisting of high- and low-energy X-ray images) needs to be first pre-processed in order to move onto higher-level analysis. This chapter considers the calibration of the acquired images and detailed approaches to segmentation. Classical approaches to segmentation in the specific context of X-ray images of a meat product are shown to be of no real use due to worthless results. A Hopfield Neural Network architecture is proposed for the segmentation of the acquired dual-band image. Performance, noise and parameter sensitivity of the fore-mentioned architecture is tested and appropriate conclusions are drawn. Next, backtracking algorithms are used successfully for the object extraction process. Then features are extracted from the obtained objects and a high-level detection is performed. Three methods for this high level detection or classification process are presented: classical, neural network and fuzzy logic based approaches. Experimental investigations are performed for all three methods.

Chapter 5 evaluates the inspection system presented in the previous chapter. A Receiver Operating Characteristic (ROC) evaluation is performed. Moreover, a

comparison between the developed system's performance and human operator's performance is presented.

The developed X-ray imaging inspection system is regarded as a generic system in Chapter 6 due to its high flexibility and applicability. Industrial X-ray imaging inspection applications using the developed system are presented: quality control for intelligent composite materials, defect inspection of welds and castings, inspection of ball grid arrays and die-attach for the electronics industry. Moreover, the applicability of the developed inspection system for classification and diagnostic tasks for medical applications is discussed.

Chapter 7 presents general conclusions of the research. Specific conclusions are drawn in the context of foreign body detection for the food industry. Moreover, further work and recommendations are proposed.

"An undefined problem has an infinite number of solutions."

Robert A. Humphrey

C H A P T E R 2

GENERAL INDUSTRIAL INSPECTION SYSTEM

2.1 Introduction

There are just a few products of industrial processes so well defined that their quality can be guaranteed to meet exactly the client specifications or requirements. However, in most areas of industrial processing some sort of testing or inspection has to be performed on intermediate or final products *[Kehoe, 1990], [Kehoe and Parker, 1992].*

Whether it is the inspection of defects in aluminium castings, detection of foreign body within food products or measurements of geometric and structural characteristics of industrial products, a basic inspection system assumes the implementation of an inspection technique. When dealing with small and simple batch products, it is easy and inexpensive to perform destructive testing on a sample of components, but in most of the industrial cases, non-destructive inspection techniques are employed. A typical industrial inspection system is depicted in Figure 2.1.

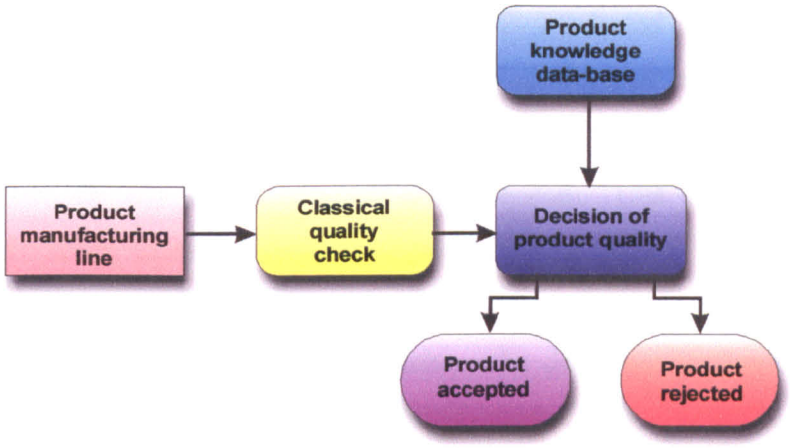


Fig.2.1 Typical industrial inspection system

From the product manufacturing line (e.g. conveyor belt) the product to be inspected is removed with the aid of some means of product manipulation and placed on a quality assurance bank. A classical quality check is performed mostly using human operators to check the main decision criteria, such as measuring physical or geometrical characteristics of the product, making sure that all components of the product are present in the final package, etc. The decision whether a final or intermediate product passes the quality or the inspection phase is taken based upon an expert knowledge database. Based on this knowledge, usually acquired over a period of time and by human experts, the product can then be rejected, considered for further processing or reprocessing, or accepted. Whatever the outcome of the inspection technique used, this process is a time consuming and a costly one. Therefore, such an inspection system has to be not only rapid, in order to assure the high-speed productivity rates necessary in a modern manufacturing process, but also to be cost efficient.

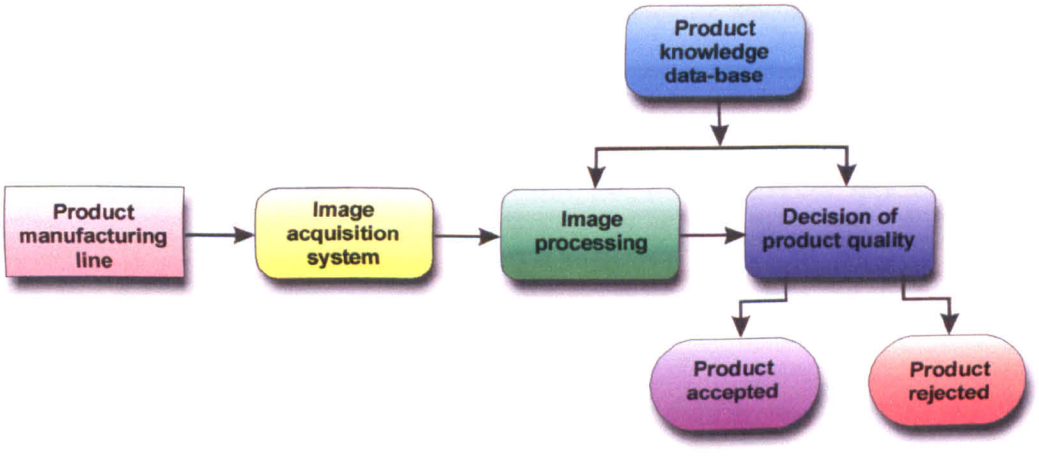


Fig.2.2 General image-based industrial inspection system

Many modern inspection systems (Figure 2.2) are based on processing an image taken from an inspected product. The image can be obtained by means of irradiating the product and using a normal or infrared television camera. Thus, an important role in this process is played by the image acquisition sub-system. This sub-system transforms physical signals taken from the inspected product into useful visual signals. Image processing techniques are then employed to enhance and analyse the resultant image and with the aid of an expert knowledge database the decision whether the product can pass the inspection tests is taken.

2.2 Specific problem and generic system

When someone takes a bite of a chicken filet sandwich, the last thing he or she wants to do is to bite onto a piece of bone in the meat. That is also the last thing the food industry wants and therefore the inspection process has to be made reliable.

Because there is an increase in demand for larger types of food products there was a need for automation of the processes involved in the preparation of the meat, fish or other food products. Using this automation system for processing the food, a higher production capacity is obtained over manual processing of the food products. One of the most important processes is the deboning one. Many fast and automated systems have been developed and are being used in the food industry to separate the meat from the bones *[Elsten, 1991], [Hazenbroek, 1991], [Jones et al., 1999], [Heidke et al., 1999], [Wols, 2000]*. However, during the automated meat processing stage, foreign bodies can remain in the meat. At a later stage the meat has to be inspected and the foreign bodies removed.

With the deboning process there is the same problem. The speed with which the meat is deboned is directly proportional to the number of fragments of bone remaining after the process. There is a trade-off between minimal meat wastage and high rejection of bone-contaminated meat. That is why the inspection stage of processed meat is very important. The meat contaminated with bones has to be rejected and then processed again. The detection of bones has to be reliable and cost efficient because if the meat is rejected and it is not actually bone-contaminated, it is sent again for deboning and that increases the production costs - this is called a "false positive".

Food processing factories have spent a great deal of time and effort trying to find accurate affordable ways to screen deboned poultry products or any other meat

products for any missed bone. Currently, most processing plants have employees who feel each piece of meat by hand for missed bone. Unfortunately, this method is not very accurate. This method still allows the tiniest pieces of bone or foreign bodies to slip by.

Many systems have been proposed and some of them are even used for detection of foreign bodies or contaminants such as bones, mineral stones, natural rubber, ferrous or non-ferrous metals, glass, etc. in the food industry. There are three main categories of inspection systems for the food industry *[Graves et al., 1998]* (Figure 2.3):

- Techniques based on measuring the difference between the weight or size of the food product and the foreign body;
- Optical inspection techniques based on sensing the colour or shape of the foreign body in comparison with the food product;
- Methods of finding embedded foreign bodies in a food product by means of irradiating it with some type of electromagnetic radiation.

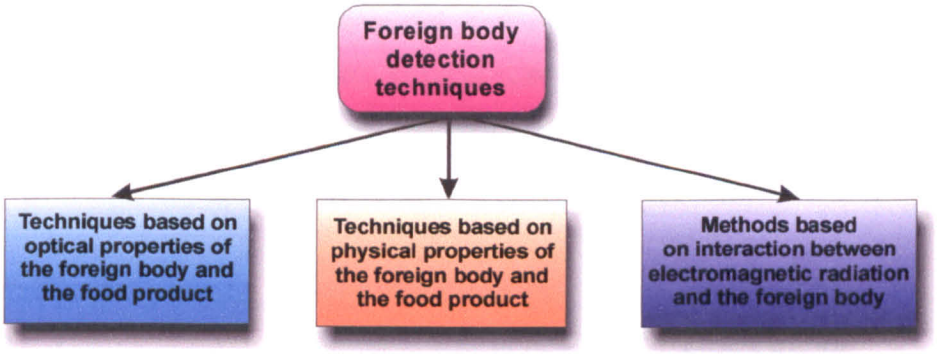


Fig.2.3 Foreign body detection techniques for the food industry

The latter methods are the most used across the food industry and they are based on the use of some kind of irradiation of the meat and deduces from the intensity of the radiation whether there is a foreign body in the meat or not. X-rays are currently used in most of those systems and it was originally proposed from medical science where an X-ray is used for detection of foreign matter in the human body.

As a consequence, systems have begun to be used in bone screening that use X-ray transmission radioscopy such as those used for security reasons in airports. The vast majority of radiographic inspection is performed using conventional radiographic equipment, consisting of an x-ray source and photographic film and a visual inspection process. A human operator examines the film images and evaluates the defects found

against a set of predefined standards. Due to the demands of high-productivity and cost efficient processing, the integration of the inspection system within a general concept of Computer Integrated Manufacturing was needed. Therefore, image intensifier with television cameras and image processing systems were introduced into the inspection process. Image intensifier systems capture radiographic images by means of fluorescent screens sensitive to X-ray radiation (see Appendix A). A TV camera monitor was added to the system and the human operator was able to carry out real-time inspection tasks by viewing the X-ray image of the product. To increase even more the speed of the inspection systems, an on-line capture system was added (as presented in Figure2.4). Thus, images were captured and transmitted directly to the computer for further processing and for automatic inspection. The acquired images need to be preprocessed in order to be enhanced and the possible noise be removed. Then, the image is segmented.

The segmentation process is the most important image processing technique employed in an image-based inspection system. It is the process of clustering or dividing the image into areas of context-specific meaning. The high-level image processing is solely depending on the segmentation results, and consequently the efficiency result of the inspection process is dependant of the segmentation process.

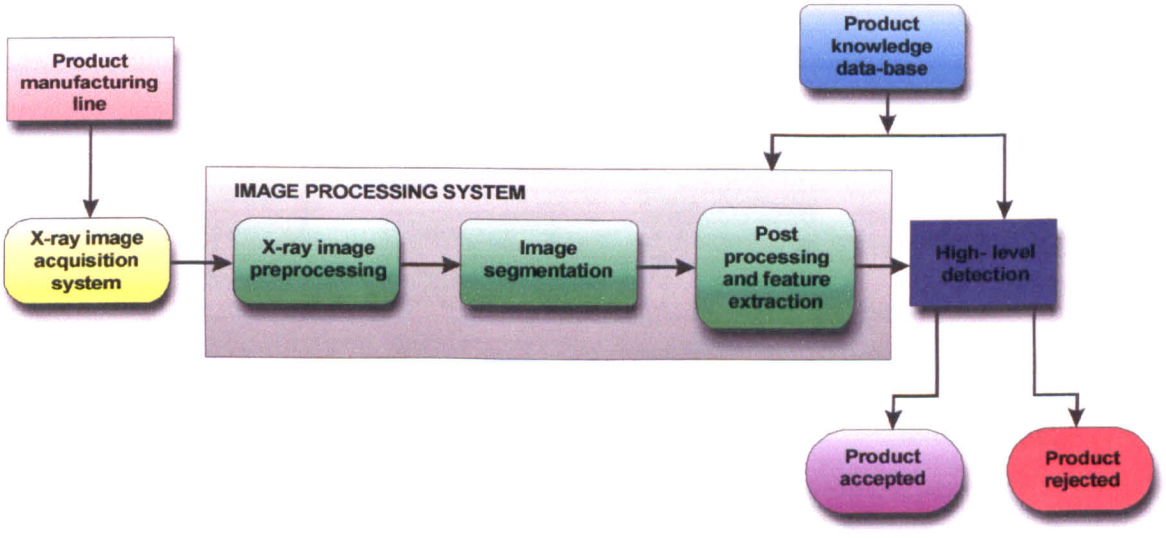


Fig. 2.4 General X-ray image-based industrial inspection system

The density difference between meat and bone is small and the variation in density of the meat and the presence of fat or water makes it difficult for X-ray

techniques to discriminate accurately. Another source of problems is that the meat has, in general, a random variable thickness, e.g. chicken bone. Because of this random thickness, the intensities of the transmitted x-rays will in general vary at different locations even though no bone is present. Also, the thickness of meat can mask the bone and prevent the system detecting it. Because the rate of false positives is very high, sometimes around 50%, an increase in production costs occurs. In the screening process of poultry pieces, for instance, it is very important to identify fragments of bone and cartilage with a high degree of accuracy and at a reasonable speed.

Thus, the automated processes have to be made reliable and time and cost efficient in order to be widely used in the food industry. "Intelligent Manufacturing Systems" – (IMS) was the first European company which made an important step in automation by including a computer system to replace the human operator. In this way, greater productivity was obtained and also a high detection rate for bones in the poultry business. Also, the number of false positives was significantly reduced as opposed to the traditional way (feeling the bones by hand).

In the following paragraphs different existing systems will be presented along with their claims and disadvantages.

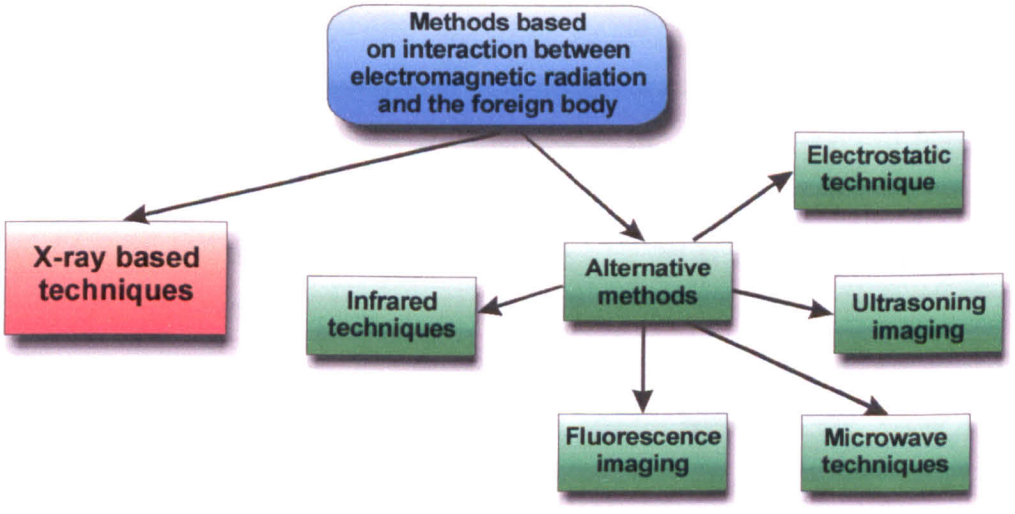


Fig.2.5 Foreign body detection techniques based on radiation

2.3 Proposed and existing systems on the market

There are only a few systems for detecting bones in meat around the world. Most of them are implemented in the USA and Japan. Many different methods for

screening food products such as mechanical detection *[Smith et al., 1971&1972]*, computer method *[Krug et al., 1996]*, laser irradiation *[Meyn, 1991]*, optical detection *[Heiland et al., 1993]* etc. are used for detection of foreign body in food products.

Since the foreign bodies present in a meat product are mostly embedded within, methods based on irradiating the product are applicable (Figure 2.5). The reason underlying these methods is the fact that the radiation can penetrate the food product and the difference between the incident radiation and the resultant radiation energy can be analysed and a decision whether a foreign body is present in the product can be taken accordingly. Among these methods, X-ray based inspection techniques are the most widely used across the food industry.

2.3.1 X-ray based methods

The majority of the systems are based on the process of irradiating the meat with X-rays, any changes in the intensity of the radiation being an indication for the presence of foreign bodies (such as bones) in the inspected meat. Because X-rays have short wavelengths and high energies, they are able to penetrate into biological materials that are not transparent to visible radiation. In most systems the transmitted X-rays are detected by an X-ray sensor and analysed either by simple threshold based techniques or by various type of image-processing algorithms. There are three types of X-rays detectors (sensors): using phosphor screens which convert the X-rays to light and then use an optical imaging system, X-ray intensifiers which comprises of the two parts mentioned before in one unit and a linear series of solid-state X-ray-sensitive elements which will transmit linear sample of information about the product

The idea of automated contaminants detection within food products goes back to *[Ramsay et al. 1976]*. They have developed a system based on the X-ray irradiation approach. As mentioned before, because of the random variation in the meat thickness, bones can be masked by meat or imaginary bones can be detected when the meat is actually clean. In order to obtain a uniform thickness for the meat product that will be screened, *[Ramsey et al. 1976]* came up with a solution to the inspection problem. Before exposing the meat to X-rays, a flowable material is placed about it, in order to form two opposite parallel surfaces. The coefficient of X-ray absorption for the flowable material should be the same as the meat or close.

The X-rays are then directed along a path through both of the formed parallel surfaces. In their experiments, they used buckets with flat bottom in which they placed the chicken meat. The buckets were then filled with water, because it has similar coefficient of absorption as the chicken meat. The X-rays were then directed through the top surface of the bucket and through the bottom surface. If there are no bones present in the meat, the intensity of the transmitted X-rays into that combination of water and chicken meat will remain almost constant in any location (with certain variations of course). If bones or foreign bodies are present in the meat, the intensity of the transmitted X-rays will suffer a sudden change. The meat is screened by passing continuously between an X-ray source and a number of X-ray detectors. The detectors are opposite the X-ray source. In this way the meat is actually partitioned in multiple areas corresponding to each detector. Each detector will be responsible for detecting the intensity of X-ray through a small portion of the meat. An electrical signal is then generated by the scanning of the meat, which is differentiated to produce a pulse at the leading edge and at the trailing edge of the bone. If this pulse has substantial amplitude, then it is used as an indication of a foreign body present in the scanned meat. An important claim of this invention is that it reduces the number of false positives. As an experiment a series of flat-bottomed containers filled with water and meat are used. Then by simply detecting the difference between the thresholds for bones for two successive containers is pretty much the same. If a significant difference is detected between those two containers, this is used as an indication of the presence of bones or foreign bodies in the meat.

Such a system *[Ramsey et al., 1976]* comprises containers for the meat, means for filling the containers with water (or other flowable substances with the same X-ray absorption coefficient as the inspected meat), an X-ray source, a number of X-ray detectors, means for moving the containers with a certain speed through the X-ray source and detectors (such as a conveyor belt), means for separating the water from the meat after the screening process and also an electrical circuitry for producing the threshold necessary in detection. Because the process of inspection has to be done in real time, the importance of the conveyor belt speed is critical. The *[Ramsay et al. 1976]* use of water acts as a delaying agent and thus the inspection system speed is considerably slow. Also the quality of the food may be affected during the covering process with the flowable substance.

Researchers at Georgia Tech completed a feasibility study on enhanced X-ray technology designed to improve the accuracy of automated bone detection by reducing false positives. Their work resulted in a USA Patent *[Papanicolopoulos et al., 1995]*. Their patent claim is for improved accuracy of X-ray bone detection systems. The concept involves combining traditional X-ray scanning with secondary scanning techniques that gather information from the X-ray scatter field based on the idea that each material produces a unique scatter pattern. Scattering is a natural part of the X-ray process. Part of the X-ray is absorbed into the material being studied (such as a piece of chicken), part passes through, and the rest bounces off in different directions, or scatters. Different materials cause different patterns of scatter (e.g. water, fat and bone will all scatter an X-ray differently). The unique features of the scatter field can help identify the materials present.

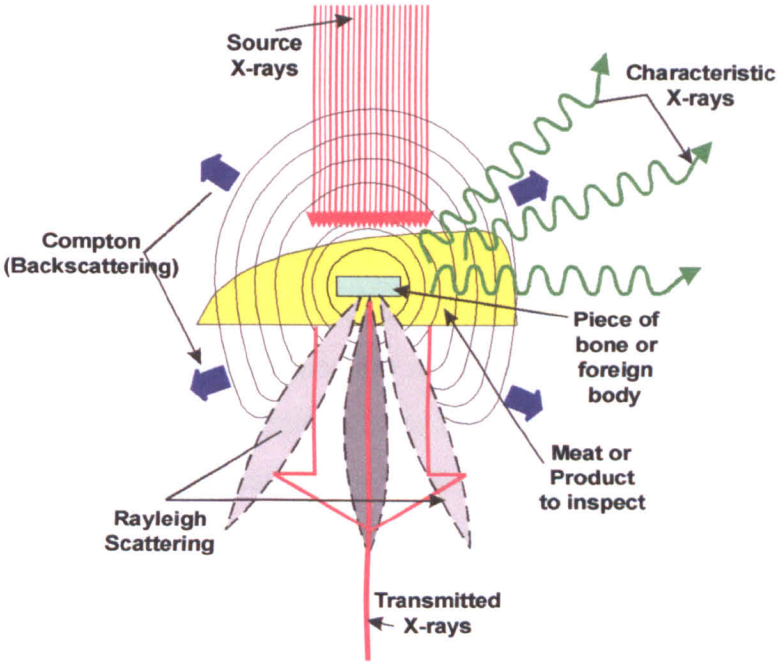


Fig. 2.6 X-ray interactions with matter

Interactions between meat with a bone within and an X-ray beam can be explained as follows (see Figure 2.6): the transmitted X-rays have a variation in intensity due to the partial absorption or blockage by the foreign body – in this case – a bone. Because it is a bone, it has a crystalline structure and scattering appears under a specific angle, depending of the object structure and composition. There are two types of scattering. One is coherent, called Rayleigh scattering, that is less pronounced for

amorphous materials (water, fat, muscle) and the second one is incoherent and is called Compton scattering, which is more pronounced for bone than for meat. Another important element is the characteristic X-rays which depend a great deal on the composition of the foreign body within the meat.

The system *[Papanicolopoulos et al., 1995]* analyses three modes of X-ray interaction for detecting the location of bone and cartilage fragments in pieces of poultry meat. The first mode is the detection of Compton scattering. This mode is used because it is sensitive to density differences between the composing elements of the meat. The second mode is detection of Rayleigh scattering. This mode is sensitive to the presence of crystalline materials, e.g. bones, and also improves contrast by clearly differentiating between crystalline material (bone) from the meat that surrounds it. The third mode is X-ray transmission just like in airport security systems. This third mode is used to determine the exact location of the foreign body fragments. It is also used to normalise the detected Compton and Rayleigh scatter fields. There is a need for normalisation because the variation in thickness of the inspected product may cause variations in intensity.

A fourth parameter is also taken into account: the ratio of Rayleigh-to-Compton (R/C) detected intensities. It is claimed that this ratio makes the detection process very accurate because the ratio for bone or cartilage is as much as five times greater than the one for meat or fat tissues. These ratios are stored in a computer for later comparison with instantaneous R/C readings which creates the potential for detecting false readings (e.g. if the Rayleigh detector becomes very noisy and its readings lead to a presence of a bone, the comparison with old values of the ratio R/C may indicate that this is actually a false positive).

The *[Papanicolopoulos et al., 1995]* system comprises an X-ray source, food conveyor, Compton scattering detectors, Rayleigh scattering detectors, X-ray transmission type detectors, a computer system or a processing unit. The X-ray source has to be of a sufficient energy and intensity to power the scatter fields and is positioned above a moving belt. The X-ray should be of a monochromatic type because using polychromatic rays will lead to a perturbation within the scatter field. Also, above the belt are positioned the Compton detectors which detect the intensity of the Compton scattering and their output is fed into the computer or processing unit. Below the belt are positioned the Rayleigh detectors which detect the intensity of Rayleigh scatter

field, and the transmission type detectors for detection of changes in intensity of the transmitted X-ray. Their outputs are also applied to the computer or processing unit. The computer or processing unit is able to pinpoint the location of "extremely small aberrations" which may be due to skin, fat, water or bones. Using the fourth parameter R/C ratio the false positives will be eliminated by comparison with known values for bones or cartilage.

Another version of the system can divide the screening process into two stages each having its own X-ray source and detecting only one of the scattering fields. One stage will detect Rayleigh scattering using low energy X-ray source (-30 KeV) and the other stage will detect Compton scattering using high-energy (-80 KeV) X-ray source. Because there are two scans in parallel and not consecutively, high speeds for the detection systems are possible.

Prior systems to *[Papanicolopoulos et al., 1995]* utilise "backscatter" detection of radiation from Compton scattering to enhance the image produced by the transmission radiation *[Friddell, 1990]*. *[Dzubay, 1976]* relies upon photon fluorescence emission by the inspected product when it is bombarded with X-ray. His method is designed for analysing samples such as compressed orchard leaves and it is not suitable for detecting bones or cartilage within chicken meat because those have low density that may cause very low fluorescent intensities. Fluorescence in the way Dzubay uses it may only detect foreign bodies on the surface of the food product. For the same reasons, the screening systems described by *[Badano et al., 1989]*, *[Page et al., 1984]*, *[Pavlik, 1969]* can never be adapted to work for detecting bones or cartilage in chicken meat.

A general X-ray inspection system is described by *[Annis et al., 1993]*. The systems presented above use X-ray to create grey-level images on a monitor or a computer screen. Because such images can be difficult to interpret by an operator, Annis et al. proposed a system that creates a colour image of the product/object that is required to be inspected. Three colours are used to display three types of material which are the most important: one colour for pixels which represent more than a predefined level of attenuation; another colour for pixels which represent more than a predetermined level of backscatter and finally another colour for the rest of the energy that is transmitted through the product/object. Such a system comprises a means for irradiating the object with penetrating radiation (X-ray radiation); a means for detecting

radiation transmitted or backscattered through and from the object and a means for displaying pixels in their appropriate colours. Due to the fact that many areas of the object will correspond to both transmitted and scattered energy, their solution may lead to an image with composite colours that can be very difficult to interpret. Therefore, in the case of detecting foreign bodies within meat products, where the meat can fold and create a random thickness to mask the contaminant, this system can prove to be very difficult to use.

Frozen fish fingers are made from a mixture of different kinds of fish that arrive at the processing plant as frozen blocks. To make the fingers, the frozen fish is minced and while still frozen it is pressed into the desired shape and other ingredients are added to it (such as batter and breadcrumbs). Then the product is stored in a deep freezer. Because the consumer of these kinds of products are typically families with children or restaurants, even the smallest bone is unacceptable. An X-ray detection system was designed and implemented by *[Kelstrup, 1998]*. His system comprises a high resolution imaging chain with a high conversion factor and sensitive optical coupling to optimise the contrast. It is claimed that the system can detect pin bones of about 0.3-0.5 mm thickness, while scanning at speeds of about 100 mm/s, in frozen blocks of the raw food material which make up a fish finger of around 60 mm thick.

A non-destructive X-ray inspection apparatus for contaminants, with applicability in the food industry, is presented by [*Ocleppo, 1999*]. This system has been designed for the inspection of glass vessels and is trying to solve a problem found in conventional systems where the shadow of the product can mask the foreign bodies present within it. The problem is solved by using two sensors placed at a 45^0 angle with respect to the inspected product (the glass vessel). Therefore a simultaneous failure of the two sensors is statistically impossible and the end-user can carry on the inspection process with only one sensor until repairs are undertaken. The apparatus consists of a conveyor belt, a standard X-ray irradiation source and two radiation detectors or sensors.

[Takahash et al., 2000] describes a contaminant-detecting apparatus having high selectivity and high sensitivity against a contaminant. The product is conveyed and irradiated with X-rays from a standard source (Figure 2.7). A two-dimensional (2D) image is created using an X-ray detector and is stored in a storage unit. The 2D image reflects the distribution of X-ray within the inspected product. A calculation unit

performs a sum-or-product operation of a kernel, using a predetermined coefficient matrix. A difference unit calculates the difference between the intensity of X-ray stored in the storage unit and the one calculated using the calculation unit. A determination unit compares the difference with some predetermined criteria, therefore being able to detect the presence or absence of a contaminant within the food product.

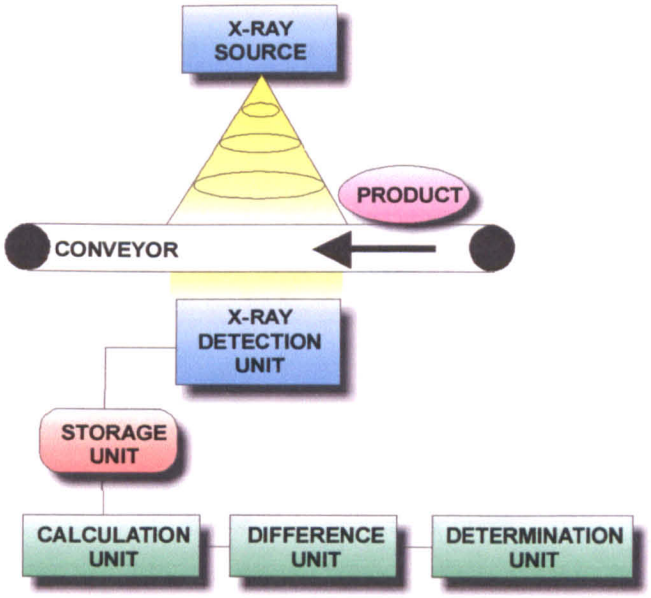


Fig.2.7 Schemata of Takahash et al. inspection system

2.3.2 Alternative methods based on radiations

As an alternative to X-ray detection Meyn has designed another method of foreign-body detection. A claim of *[Meyn, 1991]* is that use of X-ray detection systems in the food industry is not commercially attractive to the consumers, because there is evidence to suggest that they do not want to read on the product label that it was treated with X-rays. This may lead to a reduction in the potential commercial success of such a food product. Meyn 1991 has designed a system that uses a laser beam instead of X-rays. It is his claim that treatment by lasers light has no negative connotations by consumers. The method consists in scanning the meat by irradiating it with a laser beam. Whenever the intensity of a local beam is changed, then it may indicate the presence of a foreign body such as a bone. The irradiation of meat with laser light has no effect whatsoever on the meat and thus does not present public health hazard [*Meyn, 1991*].

Such a system comprises a conveyor for the food products, a scanning unit, laser light source, means for detecting and measuring intensity variations in the emitted laser beam, means of intercepting each emitted laser beam, a computer system or an electronic device capable of making the decision as to the presence or not of a foreign body in the food product, as depicted in Figure 2.8.

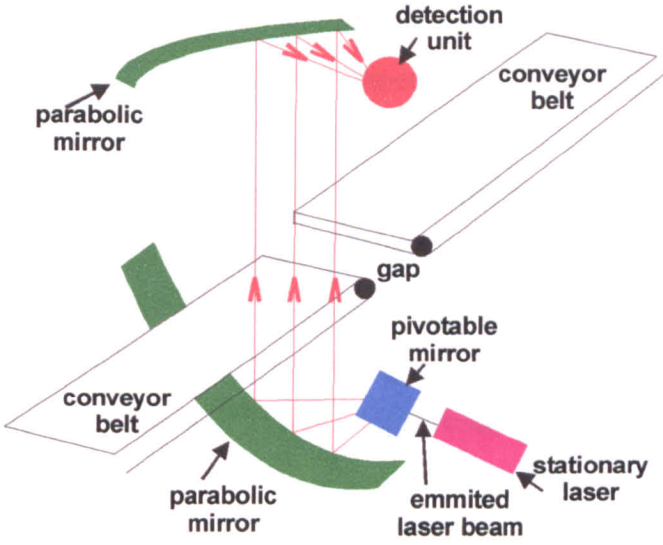


Fig.2.8 Laser irradiation *[Meyn 1991]*

The intensity of the laser beam through the irradiated meat is proportional to the density of the meat, or if foreign bodies are present in it proportional to the undesired ingredients contained. The detected intensity of the laser beam is then given to the computer that makes a decision concerning the presence of undesired ingredients or not. The computer also has to know the wavelength of the laser beam, the type of meat that needs to be scanned and the type of ingredients that are expected to be found in it. The system scans the product line by line and the results of the local intensity are fed into the computer. The computer creates a map of the local intensity associated with the meat product (a computer image). The scanning part of the system comprises two parabolic mirrors positioned above and below the food product conveyor and a laser source that is movable along the surface of one of the mirrors. These positions make the scanning possible by allowing one laser beam to cover a plane that intersects the food product (line by line scanning).

Ultrasonic imaging also plays a role in foreign body detection systems. The ultrasound has the ability to propagate through biological materials. At the area that surrounds a foreign body within the food product, the ultrasound is partially transmitted

and partially reflected. This phenomenon can be used to indicate the presence of a foreign body. There are three methods available: a single point measurement, a slice through the product and time gating of the signals such that information from a certain depth is measured [*Lawson, 1996*]. The ultrasound method has a major shortcoming, because the food product and the ultrasound transducer need to be acoustically coupled. Thus the product is often immersed in an acoustic medium such as water. This method has been shown to be more suitable for the inspection of vegetables such as potatoes or apples as mentioned by [*Graves, 1998*].

Another possibility of detection is by use of *microwave techniques*. If microwaves pass through food materials containing moisture in the form of water, the energy is strongly reflected by the water molecules [*Graves, 1998*]. The disadvantage of this method is that it is very difficult to distinguish between a high-density sample of high moisture content and a low-density low moisture sample.

An *electrostatic technique* is also possible to detect the presence of contaminants within food products. The method is based on the property that the capacity of a parallel-plate capacitor increases when a dielectric material replaces the airspace between the plates. By making the food product pass through the plates of a parallel-plate capacitor, an image can be built for that product [*Faruq, 1991*]. This method was implemented by a company in Huntingdon, UK – (Laetus Ltd). mostly for non-foreign body detection such as content verification or crack detection of the inspected packs.

Another system uses *fluorescent lights* for detecting the bones or foreign bodies inside food substances *[Koch et al., 1998]*. This system has two belt conveyors. The first conveyor is constructed entirely from plastic material. A number of fluorescent lights are positioned below the first conveyor. The second conveyor is placed above the first one in such a way that the inspected product (chicken meat) will pass between the two conveyors. In this mode, the meat is forced to have low thickness because, the light will not reveal anything if the parts are too thick. Originally, the system was used by a human operator who, through rose glasses observes the product for contrast. A video camera can also be positioned inside looking down on the flattened parts. It allows the images to be fed into a computer, where via image recognition techniques the presence of foreign bodies within the meat can be detected.

2.3.3 Commercially available systems for food inspection

Commercially available systems for food inspection are also mainly based on X-ray irradiation of the inspected product. A review of available systems is presented in the following paragraphs.

One of the first systems available on the market was BoneScan designed and manufactured by Spectral Fusion Technologies (formerly IMS) Ltd., UK. This is an on-line inspection system capable of detecting and automatically rejecting food products containing foreign bodies. SFT claims that their system has a detection rate approaching 98% with only a 3% false rejection rate. The SFT system is claimed to be capable of addressing other problematic issues such as random thickness of the meat, different breeds of poultry and variations in the X-ray sensor due to changes in the ambient temperature.

Another inspection system has been developed by CINTEX Ltd, UK. The CINTEX INSIGHT X-Ray system it is claimed "*provides an efficient, cost-effective method of detecting contaminated, missing and even misaligned package contents directly on the production line*". Their system also solved the problem of potential instability of the system with the ambient temperature and also minimises the need for calibration of the machine.

LOMA's range of X-ray food inspection systems (AXIS) are designed to minimise the contaminant risk to workers, being able at the same time to detect glass, stone, calcified bone, high density plastics and rubber. Their system family is built with two variable power-bands – high-power and low power for an X-ray detecting system that allows the end-users to select the most suitable one for their own needs.

BARCO also present X-ray inspection systems for the detection of high-density defects such as bones in meat, poultry or fish, metal or high-density plastic in packaged goods, stones in bulk products such as nuts or beans, and glass or metal fragments in jars or cans.

HEIMANN Systems – USA offers X-ray inspection systems for food products along with other type of X-ray vision systems such as baggage inspection systems, or truck and van inspection systems. Their family of X-ray machines allows the end-user to inspect bulk products, individual packages, products in cases and cans and jars.

Another British company, SAFELINE has introduced a new method of inspection for metal contaminants – the use of a low power, high frequency magnetic

coil system, together with latest digital signal processing. The principle of operation is simple and is based on the fact that the magnetic field is disturbed by the presence of metal contaminants within food products. This disturbance produces electric signal that is interpreted by digital electronics. The system proposed by Safeline can be used as in-line or end-of-line inspection machine for the detection of metal contaminants within finished, packed and sealed goods whether it is packaged within aluminium foil, metallic film, tin, cans or jars. Typical applications for in-line pipeline metal detection and rejection systems include a host of varied food products such as soups, baby food, pickles, sausage meat, sauces, peanut butter, ice-cream and liquid chocolate.

PARTECH Vision from the US use X-ray techniques in their inspection systems – QScan line, these are specially designed for searching out contaminants such as glass, metal, stones, bone and other foreign objects inside filled and capped containers such as glass jars, metal cans, boxes and even aluminium foil or plastic pouches.

The GRASEBY group offers X-ray based systems (CDX line) capable of automatically detecting and rejecting manufactured products contaminated with glass, stones, bones, metal, plastics, and other foreign dense materials. The CDX systems can also be used in soup and slurry applications for contaminant detection prior to packaging.

A German company, Fraunhofer IIS-A, is manufacturing a system for X-ray radioscopic image processing – ISAR. Even though, ISAR is designed for the detection of pores and blowholes in x-ray images of metal products, it can be adapted for the detection of contaminants in food products.

All above mentioned commercially available systems use a Graphical User Interfaces (GUIs) in order to achieve speed and ease of use by the end-users (operators). Most of the systems use a standard computer monitor or touch-screen, which ensures that relatively unskilled operators can understand and easily work the inspection systems.

A summarised list of the products and type of product defect and typical contaminants found in the food industry are presented in Table 2.1. Table 2.2 shows a short summary of the existing manufacturing companies and their system usability with respect to food products. As can be seen, the majority of automated systems are used for the inspection of meat, fish and poultry products. There is no universal system for the

inspection of all types of food products. However, some commercial systems have a large range of applicability, but they require specific training and calibration.

Table 2.1 – Types of defects and contaminants in the food industry

Product/ defects or foreign body	Packaged Products	Cooked Products	Raw Meat (Poultry, Fish)	Eggs, fruits	Cans, jars	Glass Vessels
Item missing in the package	√	√		√		√
Spice packets missing	√	√				
Package missing in carton	√	√	√	√	√	√
Box or bag missing from case	√	√	√	√	√	
Wrong size of the product	√	√	√	√	√	
Sauce packet pierced inside the product		√				
Product too thick/thin		√	√	√		√
Product too light/heavy	√	√	√	√	√	√
Case deformed	√	√	√	√	√	√
Case pierced	√	√	√	√	√	
Film lid pierced		√				
Non-ferrous metals shreads present within product		√	√	√	√	√
Metallic shreads present within product		√	√	√	√	√
Glass present within product		√	√	√	√	√
Bone present within product		√	√	√	√	√
Stones present within product		√	√	√	√	√
Plastics (PVC, rubber, plastic film) present within product		√	√	√	√	√
Metallic trays deformed	√	√			√	
Water present in the product	√	√	√	√	√	√
Product too salty		√	√		√	√
Change in temperature of the product		√			√	
Product overcooked		√			√	
Change in colour of the product		√	√	√	√	√

Table 2.2 – Inspection companies and type of products

Product	SFT	LOMA	CINTEX	BARCO	HEIMANN	SAFELINE	PAR	GRASEBY
Meat, fish and poultry	√	√	√	√	√			√
Tins, jars and cans			√	√		√	√	
Cooked products	√	√	√		√		√	
Fruits, eggs and vegetable		√	√	√	√	√		
Glass vessels			√				√	
Packaged products		√		√		√	√	
Bakery		√			√			
Cereals		√	√		√	√		
Dairy		√				√		
Soup and slurry			√					√

2.4 Summary and recommendations

The majority of the systems used at present still require a human operator. In *[Koch et al., 1998]* the operator is necessary to read the video image of the inspected product in order to make the decision of acceptance or rejection. Another disadvantage of these systems is that they are only able to reject the product and not able to pinpoint the location of the foreign body, see *[Ramsay et al., 1976]*, *[Meyn, 1991]*, *[Kelstrup, 1998]*. Those disadvantages can be overcome by using a computer system or a processing unit. The role of the computer is to analyse the data and make the decision. In Koch's system, a video camera is used to take images of the inspected product and a computer is used to identify and reject parts with bone fragments found by image recognition techniques. A computer system is also used in *[Papanicolopoulos et al.,1995]* system for normalising the data obtained and making a comparison between new values and stored, known, old values relating to the previous detection of foreign bodies within chicken meat.

Human operators usually suffer from fatigue and subjectivity when dealing with inspection tasks and their performance, as quality inspector tends to decrease over time. Usually operators are given regular breaks and this can lead to gaps in the production process and consequently incurring a cost.

Therefore, a modern inspection process has preferably to be undertaken without human intervention, or at least with as little human decision making as possible. One

cannot talk about subjectivity or tiredness, as humans perceive it, of a computer system as an integral part of an inspection system. Thus, a computer system has to make the correct classification of the inspected product and also have to perform all necessary pre-processing computations.

The scope of the thesis is to explore novel approaches for an reliable and robust automated computer-based foreign body detection system for the use in the food industry that minimises human intervention.

"When you steal from one author, it's plagiarism; if you steal from many it's research."

Wilson Mizner
1876-1933

CHAPTER 3

IMAGE PROCESSING- A LITERATURE REVIEW

3.1 Introduction

As shown in the previous chapter, an important function of an image-based inspection system is the image processing stage. A system that has the objective to minimise human intervention and is one that uses an image as the input and take that through the consequent analysis stages requires that image to be automatically translated into meaningful information. This method is called image processing.

Most authors agree that image processing is used for two major purposes:

1. to improve the visual appearance of images to human interpretation, and
2. processing of scene data for autonomous machine perception

The techniques used for each of these tasks is not always the same, but there is considerable overlap. Image enhancement and restoration procedures (1 above) allow scientists from various areas of activity (physics, archaeology, medicine, biology, law enforcement, defence and industry), to process degraded images, to reconstruct lost artefacts and to provide a better quality of image for humans to interpret.

The purpose, of 2 is to extract information in a suitable form such that it can be analysed and used by computers. Typical areas of use are character recognition, face recognition, automated fingerprint processing, industrial machine vision, industrial inspection methods, automated processing of satellite images and processing of biomedical images.

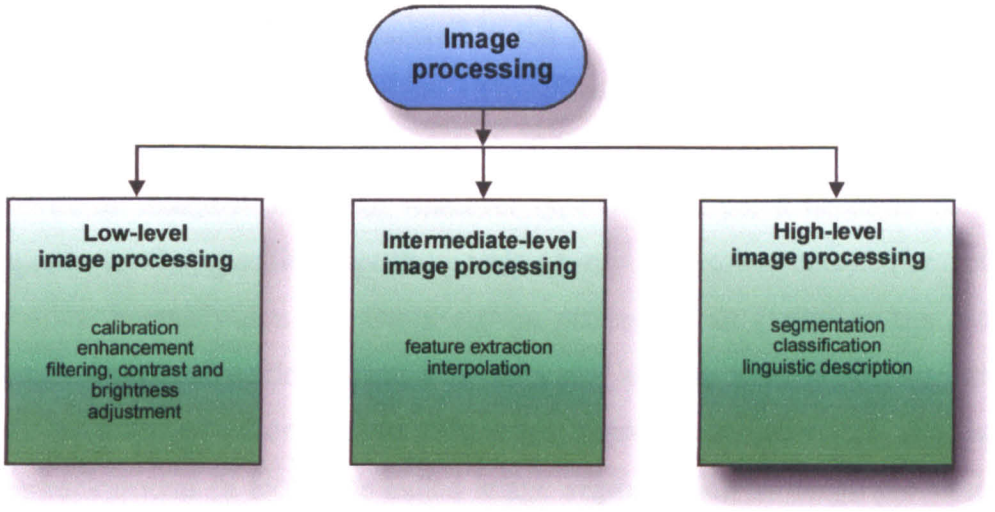


Fig. 3.1 Image processing techniques

A taxonomy of image processing techniques is presented in Figure 3.1. At the lower level, there are techniques for the calibration and the enhancement of corrupted images; at the intermediate-level feature extraction and interpolation are among the most well known methods, and in the high-level, segmentation, classification and linguistic description of images play a very important role. The low-level image processing techniques are sometimes referred to as pre-processing algorithms and techniques, whereas some authors consider the intermediate-level methods as post-processing techniques.

Any image can in general be described by a two-dimensional function $f(x,y)$, where x and y represent the spatial coordinates and $f(x,y)$ the value at that location. Depending on the type of image, the value $f(x,y)$ can be either light intensity, temperature (for thermal images), intensity of X-rays (for X-ray images), intensity of radio-waves (for nuclear magnetic resonance images – MRI), depth for range images, and other.

A digital image can be viewed as a two-dimensional matrix. Each element of the matrix is called a pixel *(x,y)* and it represents a discrete value of the feature intensity value:

$$I = \bigcup_{\substack{x=1,m \\ y=1,n}} f(x,y), \tag{3.1}$$

where m and n are the dimensions of the image.

Most of the techniques presented in the following paragraphs are designed and implemented for light intensity images. Therefore, in the subsequent paragraphs we shall mainly refer to $f(x,y)$ as the 'grey value' of the pixel at location (x,y).

Once an image is acquired, whatever type it might be, it needs to be analysed with regards to the specific inspection task (X-ray inspection here). The process is similar to the human vision and interpretation process of an image. In the process of understanding the real world humans rely on senses and the most important sense used is vision. A human operator extracts meaningful objects from a scene, separating them from the background (a technique called image segmentation in image processing) and consequently a decision is taken. Only then, are the extracted objects analysed in order to understand the particular scene. Thus, image segmentation is the crucial image processing technique used as the core of an automated image-based X-ray foreign body detection system for the food industry. X-ray images from an image acquisition system need to be partitioned or segmented into objects of meaningful importance, according to specific criteria, present case considered in this thesis being the segmentation of chicken breast meat X-ray images.

3.2 Image segmentation

Image segmentation is one of the most important processes in modern computer vision. It involves partitioning the image into meaningful segments. It is the process by which a computation translates the original image description i.e. an array of grey levels – into segments with uniform and homogenous characteristics. They should correspond to structural units ('objects') in the scene. A typical example would be extracting objects from a complex background.

Thus, using (3.1), the segmentation can be described as partitioning the image I into $S_1, S_2, \ldots, S_n$ segments of objects so that

$$I = \bigcup_{i=1}^{n} S_i \tag{3.2}$$

$$\text{and } S_i \bigcap_{i \neq j} S_j = \Phi, \tag{3.3}$$

where n is the number of objects of interest.

There are many applications of image segmentation. From the medical field to robotics, image segmentation has played, and plays, an important role. For instance, for the automated detection of cancerous cells from mammographic images, segmentation followed by recognition or classification is required. Another example is that of automatic non-destructive testing techniques, such as automatic inspection of welding, castings and the detection of foreign bodies within food products. *[Lawson, 1996], [Amza et al., 2000], [Graves, 1999].* Such techniques involve the segmentation of the image, and the detection (recognition) of possible anomalies, or foreign bodies, within. Therefore the output of such a system is, in most cases, directly dependant on the segmented output of the original image.

Despite a rich literature and various methods of implementation, image segmentation remains a difficult problem due to the lack of a general mathematical model. A comprehensive number of techniques have been proposed in the literature and are briefly described below in the thesis. Selection of one method over another solely depends on the type of image that needs to be segmented.

There is no universal method that can be successfully applied to all types of images. Moreover, even the process of choosing the right segmentation method is a very difficult process. Methods developed for one type of image might not be applicable to different images, or if chosen, applied with low performances.

Segmentation techniques can be placed into three classes (Figure 3.2):

- Classical algorithms, mostly based on mathematical or statistical methods
- Artificial Intelligence based techniques
- Other techniques which either crossover or fall into none of the first two categories.

The classical algorithms include characteristic histogram thresholding, edge/boundary detection, region extraction or region growing, relaxation, semantic and syntactic approaches. Segmentation methods deriving from the artificial intelligence field such as artificial neural networks (ANNs) and fuzzy approaches can also be found.

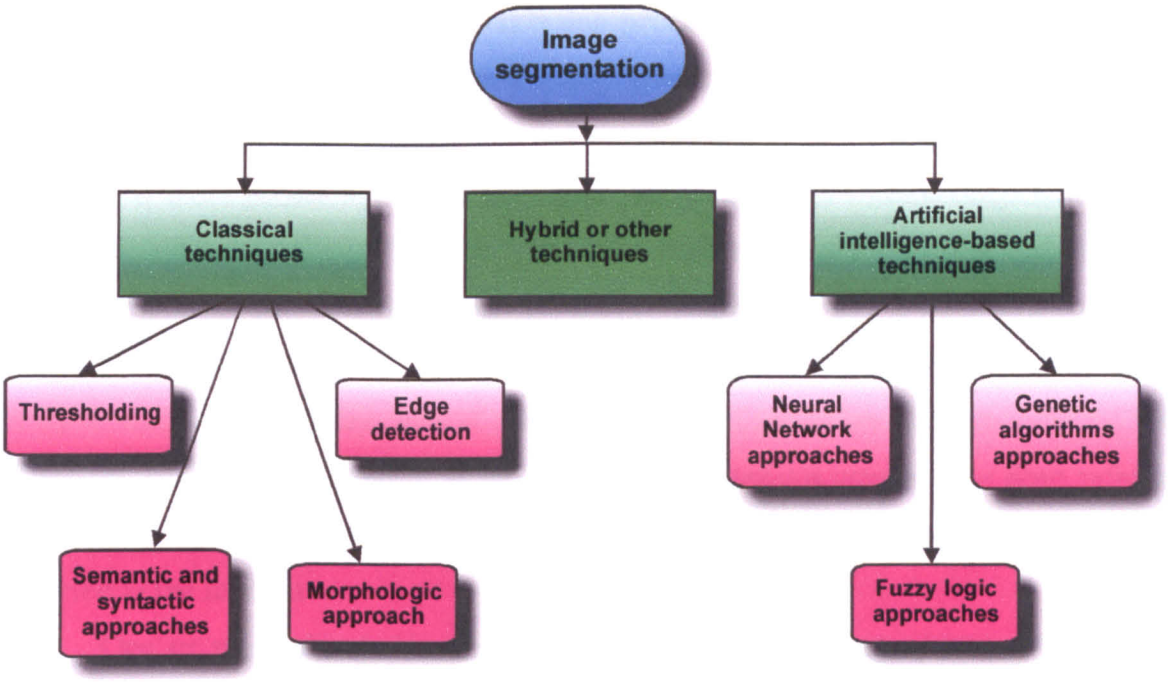


Fig. 3.2 Taxonomy of image segmentation techniques

Previous image segmentation reviews *[Fu and Mui, 1981]*, *[Haralick and Shapiro, 1985]* and *[Sahoo et al, 1988]* only surveyed classical techniques; neural network techniques have not been considered in these. A brief review of ANNs methods used for segmentation is given in *[Pal and Pal, 1993]* and *[Bezdek et al., 1993]*. Pal and Pal classified image segmentation techniques as: a) grey level thresholding; b) iterative pixel classification; c) surface based segmentation; d) segmentation of colour images e) edge detection. *[Pal and Pal, 1993]* reported neural networks based approaches used only for iterative pixel classification. *[Bezdek et al., 1993]* and *[Clarke et al., 1995]* reviewed only FFNN-based techniques used for segmentation of MRIs.

The present review of techniques is intended to be a more comprehensive study of the existing segmentation techniques, with the accent on novel artificial intelligence-based methods. Due to the enormous number of segmentation techniques reported in the literature, this survey is a selective one. Because most of the methods in the literature can be applied or extended easily to colour images and the segmentation of colour images is beyond the scope of this thesis, in the following paragraphs, segmentation methods applied only to grey level images will be presented.

3.2.1 Classical Approaches to segmentation process

Under the classical approach there are segmentation techniques based on histogram thresholding, edge detection, and semantic and syntactic approaches.

The most simple and popular segmentation method is thresholding. It is based on finding a cut-off point or threshold for the histogram of one image. The histogram of the image usually shows different peaks, each corresponding to a distinct region in the image. If the image has only one distinct object and the background, the histogram is likely to have a deep valley. In this case the bottom of the valley is taken as being the threshold. However, in practice, an image histogram does not usually have deep valleys, therefore choosing a threshold is not an easy task to achieve. There are several methods reported in the literature.

One or more thresholds can be used for the entire image *[Otsu, 1979]*, thus global or local thresholding techniques are available. Some authors divide thresholding techniques into bilevel or multilevel thresholding *[Yanowitz et al., 1989].* The bilevel thresholding *[Otsu, 1979], [Kapur et al., 1985]* assures the partitioning of the image into two distinct regions (usually black for the object and white for the background) as it is depicted in Figure 3.3. Otsu's method is based on maximizing the separability of the resultant classes in grey levels and is based on the zeroth- and the first-order cumulative moments of the grey-level histogram. On the other hand, *[Kapur et al, 1985]* assumes two probability distributions of the original image and the threshold is extracted by maximizing the total entropy of the image.

Fig. 3.3 Simple bilevel thresholding for Lena sample image

Multilevel thresholding *[Cheriet et al, 1998], [Glasbey, 1993], [Papamarkos and Gatos, 1994]* assumes that the image contains several objects with different surface characteristics. Therefore a need for multiple thresholds t_1, t_2, ...,t_k so that all pixels with $f(x,y)$ in $[t_i, t_{i+1}]$, $i=0,1,2,...,k$ constitute an object or partition. Figure 3.4 illustrates not only differences between bilevel and multilevel thresholding, but also differences of the segmentation result when different threshold values are employed. For the segmentation into three classes (case d , e and f) three thresholds were chosen. Dependant on what values were taken by the thresholds, the result presented in Figure 3.4 e is virtually unusable, whereas, Figure 3.4 d and f shows useful segmentation results.

Several studies can be found in the literature for the successful use of Otsu's method, both for bi- and multilevel thresholding*: [Kim, T,-H., et al., 1999], [Asari, et al., 1999], [Yin, 1999], [Cheriet et al, 1998].* All these approaches have a common disadvantage: they all make use of only the histogram information offered by the image, in some cases this may not be enough. Spatial information can also be used in thresholding schemes *[Weszka and Rosenfeld, 1978], [Deravi and Pal, 1983].*

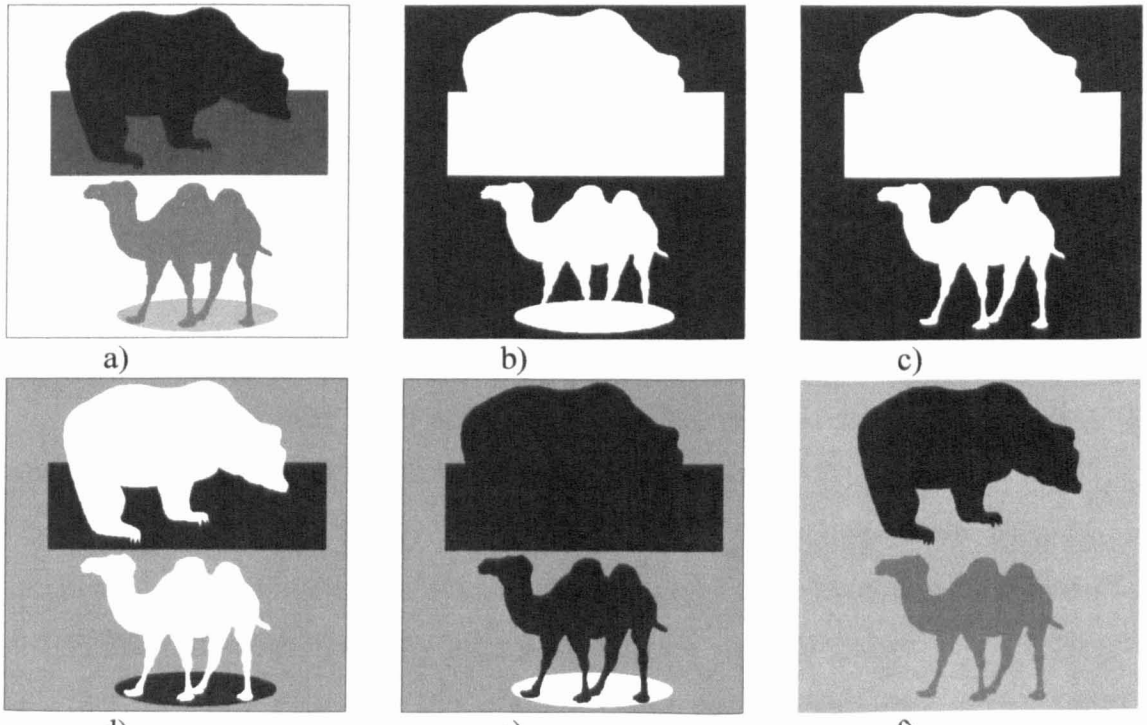


Fig.3.4 Examples of bi- and multilevel thresholding for a test image

a) original image; b) bilevel thresholding (t=125); c) bilevel thresholding (t=180); d), e) ,f) multilevel thresholding into 3 classes with different results

Thresholding can also be seen as a classification problem, where bilevel thresholding assumes classification of pixels into two categories or classes: object and background; and multilevel thresholding assumes classification into more than two classes.

Due to the fact that choosing a threshold may not always solve the problem on many occasions (when noise is present in the image or the illumination of the background is not even), adaptive thresholding techniques appeared. In such cases the objects are still lighter or brighter than the background but more in some regions of the image than in others. Therefore a single threshold for the entire image is not sufficient. Thus, the original image is divided into non-contiguous blocks and for each of these blocks a threshold is selected by using the methods mentioned above *[Chow and Kaneko, 1972], [Yanowitz and Bruckstein, 1989].*

Relaxation *[Rosenfeld and Kak, 1982]* is an iterative method to segmentation in which the decision about each pixel is taken in parallel. A decision is taken for the current iteration and it is then used for the next iteration.

Edge detection of different regions within one image can also provide a solution for segmentation. Radical changes in grey-level intensity are located and a decision whether an edge is present is taken. The edge detection method can be divided into two main categories: sequential and parallel *[Pal and Pal, 1993].* In the sequential method the decision whether a point is an edge or not is taken based upon the previous iteration for others pixels, whereas in the parallel technique, the decision is taken based upon the current pixel and some of its neighbouring pixels. *[Chien and Fu, 1974]* used this technique to detect cardiac and lung boundaries in chest X-ray images using a sequential search technique.

Classical parallel differential operators that respond to changes in grey-level were developed, such as the Roberts gradient, Sobel gradient, Prewitt gradient and the Laplacian operator *[Gonzalez and Wintz, 1987].* These operators can be used successfully in a broad range of images, for edge detection or even edge enhancement.

Morphological operators were reported to be used as a means for segmentation of a multiple range of images. The theory behind morphological operators is based on the mathematical concepts of shape. The morphology deals with two images: the original image that needs to be segmented and a basic structuring element. Each structuring element has a shape that can be regarded as the parameter of the operation.

[Thomas and Peters, 1991] reported the use of such operators for the automatic segmentation of foetal ultrasound images. Foetal femur length measurements are used to determine the age of the foetus by comparison with a typical growing chart. *A priori* knowledge of the shape and size of the femur is employed by using morphological operators.

Morphological operators were also used as edge detection segmentation techniques. A multi-scale morphologic edge detector is proposed by *[Chanda et al., 1998]*. Such an edge detector operator shows better noise immunity and orientational and positional response compared to most of the conventional methods, while being computationally less expensive.

Extraction and labelling of coronary arterial trees from single-view angiograms by the use of morphological knowledge is shown in *[Haris et al., 1999]*. Structural and quantitative information (skeleton and borders) for the artery dimensions is compared to a given artery model represented by a graph. The algorithm assumes four stages: coronary arterial tree detection, artery skeleton and border estimation, feature graph creation and artery labelling by graph matching.

In summary, classical approaches are mostly based on finding an optimal threshold or thresholds for the images, methods that are context-sensitive. They might work to solve a specific problem, but generalization is not possible. Furthermore, they can be applied successfully only when the image histogram exhibits specific local characteristics (such as similar objects are created using the grey-level from adjacent regions of the histogram; objects containing grey-level values scattered all over the histogram will lead to erroneous results).

3.2.2 Artificial Neural Network (ANN) techniques

Since 1990, artificial neural networks have come to be used as a different approach for image segmentation by iterative pixel classification. Their properties, such as graceful degradation in the presence of noise, their ability to be used in real-time applications and the ease of implementing them with VLSI processors, led to a boom of ANN-based methods for segmentation. Almost all types of neural networks have been applied with different degrees of success, the most used being Kohonen and Hopfield ANNs. In the present review, methods based on feed-forward back-propagation neural

networks (BPNN or FFNN or MLFF or MLP), Kohonen self-organising maps (SOMs), Hopfield neural networks (HNN), constraint satisfaction neural networks (CSNN), oscillatory networks (LEGION) and pulse-couple neural networks (PCNN) are presented.

An artificial neural network is a simulation of a real nervous system. It consists of a number of neurons that communicate with each other. This artificial computational model of a real nervous system was proposed in 1943 by McCulloch and Pitts. The scope of the thesis is beyond the description of ANNs in detail. More details about Neural Networks are presented in *[Zurada, 1992], [Kung, 1993], [Haykin, 1994]* or in the referenced papers.

The NN-based image segmentation techniques reported in the literature can mainly be divided into two categories: supervised and unsupervised methods (see Figure 3.5).

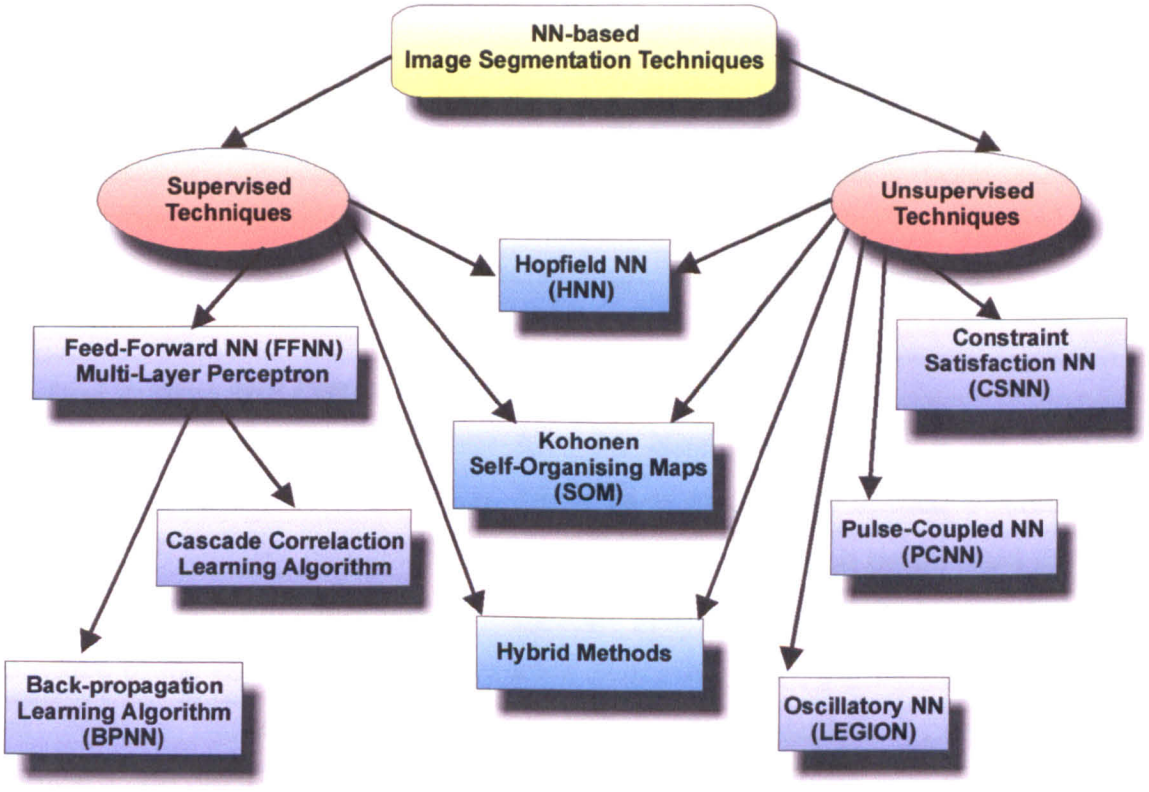


Fig.3.5 NN-based image segmentation techniques

Supervised methods require expert human input for segmentation. Usually this means that human experts are carefully selecting the training data that is then used to segment the images. Unsupervised methods or clustering processes are partially or fully

automatic. User intervention might be necessary at some point in the process to improve performance of the methods, but the results should be more or less human independent. An unsupervised segmentation method automatically partitions an image without operator intervention. However, these architectures might be implemented using application specific *a priori* knowledge at design time, i.e. anatomical, physical or biological knowledge.

3.2.2.1 Supervised Techniques

Supervised segmentation techniques (Figure 3.6) are based on human or operator knowledge to select training images and manually segment them into *k* regions. Each region is assigned with a label and the proposed architecture is trained using the selected images as training data. The method is then able to segment similar images. Labels are assigned to the regions according to the knowledge stored in the NN architecture used.

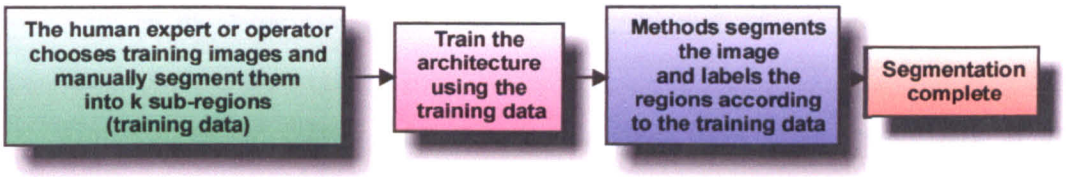


Fig.3.6 Supervised NN-based image segmentation techniques

3.2.2.1.1 Multi-Layer Perceptron (MLP)

The simplest NN architecture used for image segmentation is the multi-layer perceptron (MLP). *[Blanz and Gish, 1990]* have successfully applied a three-layered perceptron to segment grey-level images. Their BPNN approach transformed the image segmentation problem into a pixel classification problem. The input vector for the ANN consisted of the vector of features extracted from every pixel, and the output vector was the vector of classes desired for segmentation. A standard back-propagation (BP) learning algorithm was chosen to train the network.

[Lawson and Parker, 1994], [Lawson, 1996] detailed a very similar approach to image segmentation of industrial radiographic images using a back-propagation algorithm. *[de Waard, 1994]* described a MLP trained to classify pixels in a text segmentation application. Another back-propagation neural network (BPNN) was implemented by *[Silverman and Noetzel, 1990]* for detection of tumours in ultrasound

medical images. MR and CT images were also segmented using BPNN *[Ozkan et al., 1990], [Oshio and Singh, 1990], [Ozkan et al., 1993].*

A supervised method based on BPNN was used by *[Raff et al., 1994]* to segment the grey matter, white matter and cerebrospinal fluid from spin-echo magnetic resonance images. The histogram of the image was used to train a BPNN. The output of the architecture was the percentage of grey or white matter in the image.

A different method was proposed by *[Babaguchi et al., 1990].* Here, a binary image is segmented by automatically selecting a threshold by a BPNN. The segmentation problem is no longer transformed into a pixel classification problem, but into an optimal thresholding selection problem. The inputs to the ANN are the histogram values and the output is the appropriate value of the threshold.

A different learning algorithm was used by *[Hall et al., 1992].* They have used fuzzy-clustering techniques and a MLP trained with a cascade correlation-learning algorithm in segmenting magnetic resonance brain images. This approach solves the problem of speed and computational overhead that BPNN suffer from. The speed is increased by as much as two times, but in the detriment of cost of acquiring training data.

[Coppini et al., 1993] described a system based on integrating *a priori* anatomical knowledge and a feed-forward BPNN for the segmentation of target structures in tomographic images and of lung nodules in standard projection radiography. Their approach was inspired by the anatomical world: three major blocks were used to segment each type of structure. The input was the retina. Then, the structures were localised by the Attention Focuser (AF) and its findings were reported to the output of the system, called the Region Finder (RF). All three blocks were implemented as BPNN, with different parameters. The same approach is described by *[Valli et al., 1998].* Even though their performance was around 95% correct pixel classification, the presence of noise in the input images was not investigated at all. Their conclusion was that *a priori* anatomical knowledge could help improve the neural-network based segmentation of medical images.

A high segmentation rate (93.6%) was obtained by *[Seppala et al, 1997].* A BPNN was used for segmentation of low resolution and low contrast medical X-ray images. The system is basically a character recognition one, where the ID labels of medical X-ray images are automatically detected and analysed.

[Wang et al., 1992] investigated a NN used for segmentation of images containing Gaussian noise. Their approach consisted of two stages: initial and refinement segmentation. The image is represented by a FFNN so that neurons with the same state at one time represent the same region (the same object/segment). An energy function is defined for the network as being the negative of the between-class variance (see *[Fukanage, 1972]*):

$$E(V) = -\sum_{k=0}^{1} p_k(\mu_k - \mu)^2 , \tag{3.4}$$

where p_k is the probability of the k^{th} region, μ_k are the mean grey-level for the k^{th} region and μ is the mean grey-level for the image. It represents the goodness of the segmentation and its minimising will lead to the initial segmentation of the image into two subsets. The refining segmentation procedure is achieved by a "self-improvement neural network in that it learns from itself, and improves itself by a retrieval procedure". Their architecture, called a multi-layer logic neural network, was able to perform the learning algorithm in one pass. Even though this method performs well in the presence of noise, it cannot be used when the difference of grey-level within one segment is high i.e. texture regions.

[Shiranita et al., 2000] proposed a BPNN for implementation of a meat-quality grading system. A three layered MLPNN was trained with a BP learning algorithm to classify pixels as "fat-pixel pattern" or "muscle-pixel pattern". A binarization was then performed on the image. For one meat sample 9 different images were taken. The results were successful, but the proposed approach has a major shortcoming of increased time overhead necessary to acquire 9 images of the sample product inspected.

3.2.2.1.2 Kohonen Neural Network

[Haring et al., 1994] developed a Kohonen neural network *[Kohonen, 1989]* for multi-scale image segmentation based on pixel classification. Each pixel was assigned differential geometrical invariants features. The features vector was fed into a Kohonen neural network obtaining a so-called 'prototypical feature patterns'. A training image was used to train the architecture. Any image similar to the training image could be then segmented by comparing the feature pattern representation of each pixel. Thus,

supervised labelling was used to classify a pixel as belonging to classes derived from the segmentation process of the training image. The algorithm was implemented and applied for MR images of the head and showed promising results.

All supervised methods described so far have a major shortcoming: they all require a training phase using a large set of sample images before run-time. In most of the practical real-time applications, including an X-ray inspection system, the gathering of sample images is an expensive, laborious and time consuming process, if not an almost impossible process.

3.2.2.2 Unsupervised techniques

The difference between a supervised and an unsupervised technique is presented in Figure 3.7. The unsupervised method is automatically segmenting the image into *k* sub-regions and then automatically assigns labels to those regions (no training phase of the architecture is required).

Fig.3.7 Unsupervised NN-based image segmentation techniques

There are many literature reports for unsupervised NN-based image segmentation techniques. Techniques based on Hopfield NN(HNN), Kohonen Self Organising Maps (SOM), Constraint Satisfaction NN (CSNN), oscillatory NN (LEGION), pulse-coupled NN(PCNN) will be presented.

Neural hypercolumn architecture was used successfully in segmenting radiographic weld images *[Gaillard et al., 1990]*. A neural hypercolumn consists of stack of nodes that can detect changes in the spatial intensity gradient from left to right. All pixels in the left-half image are viewed as inhibitory, whereas pixels on the right are viewed as excitatory. Changes in the intensity gradient at the boundary are detected by activation of the node. Classical techniques for edge detection, such as Sobel filters or other edge-detection techniques can be implemented using such "sophisticated" artificial hypercolumn architectures.

3.2.2.2.1 Hopfield ANN (HNN)

HNN was proposed in 1985 by Hopfield as a way of solving optimisation problems. In a HNN each neuron is linked to another and weights are symmetrical, i.e. $w_{ij}=w_{ji}$, where w_{ij} represent the weight of connection between neuron i and j. The network for the optimisation application tends to relax into stable states that minimises an energy function of the form *[Hopfield, 1982], [Hopfield, 1984], [Hopfield, 1985]:*

$$E = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} v_i v_j - \sum_{i=1}^{N} I_i v_i \ , \tag{3.5}$$

where N is the number of neurons, v_i is the output of the i^{th} neuron , and I_i is the external input for the i^{th} neuron term. Hopfield demonstrated that HNN relax into a stable state tending to minimise its corresponding energy function.

HNN has been successfully applied to optimise the segmentation process where the problem is viewed as a clustering one as in *[Kamgar-Parsi et al., 1990]*. Therefore the problem can be rewritten as follows: partitioning a set of N patterns (in this case grey levels) into k clusters or classes (objects of importance) in a way that those in a given cluster are more similar to each other than the rest.

The strategy used by the majority of the authors comprises two steps: first some means of finding a binary representation for the segmentation solution, so that they can be mapped into a HNN stable state; and secondly, the definition of the energy function whose minimisation will lead to an optimum solution to the problem. Such an energy function must comprise terms for image segmentation constraints $E_{constraints}$ or syntax energy i.e. to ensure that no grey-level or pixel can belong to two classes in the same time, and terms for goodness of segmentation, $E_{goodness}$ or the semantics energy:

$$E = E_{constraints} + E_{goodness} \tag{3.6}$$

One of the first attempts to use a HNN for the segmentation problem was *[Gosh et al, 1991]*. In this approach, each pixel corresponds to a neuron in the HNN and the associated energy of the network consists of the Maximum a'posteriori probability estimate of a scene based on a noise-corrupted realisation. A possible hardware implementation of a single neuron of this approach is also presented. Noise presence in

the original images is investigated and the technique is found to be highly robust and immune to noise. The drawback of the proposed method was the fact that the images used were binary instead of grey-level images.

[Amartur et al., 1992] took the research further and proposed a HNN with winner-take-all (WTA) neurons for segmentation of MRIs. A grid of $N \times k$ neurons was used, where N is the number of pixels in the image and k is the predefined number of classes or regions. The use of T2-weighted and density-weighted MR images of the head allows the network to minimise an energy function of form:

$$E = \frac{1}{2}\sum_{k=1}^{N}\sum_{l=1}^{N} R_{kl} V_{kl}^2 , \tag{3.7}$$

where R_{kl} is the distance measure between the k^{th} pixel and the centroid of class l and V_{kl} is the output of the k^{th} neuron for class l.

A similar approach was taken by *[Cheng et al., 1996]* for the segmentation of medical images. They have used a modified version of HNN with integrated Winner Take All (WTA) learning mechanism called Competitive Hopfield neural network (CHNN). The structure of the CHNN used is independent of the image size being dependent only by the number of predefined classes and the number of existing grey-levels, as shown in Figure 3.8. In this case the energy function that needs to be minimised is defined as the mean of squared distance measures of the grey levels within each class. The Lyapunov-based energy function used is:

$$E = \sum_{x=1}^{n}\sum_{y=1}^{n}\sum_{i=1}^{k} \frac{1}{\sum_{y=1}^{n} h_y v_{yi}} v_{xi} d_{xy} h_y v_{yi} \tag{3.8}$$

where n is the number of grey levels in the image, k is the number of predefined classes or segments, d_{xy} is the square of the Euclidian distance measure between the grey-level pairs x and y and h_x is the number of pixel at the grey-level x. The performances of CHNN and Hard C-means Algorithm were compared on both computer simulated, CT and MR images. The metrics used for comparison were simple and a more appropriate

comparison needed to be done. Furthermore, the computational level of the CHNN is high, and there is no guarantee that the network does not settle down to local minima.

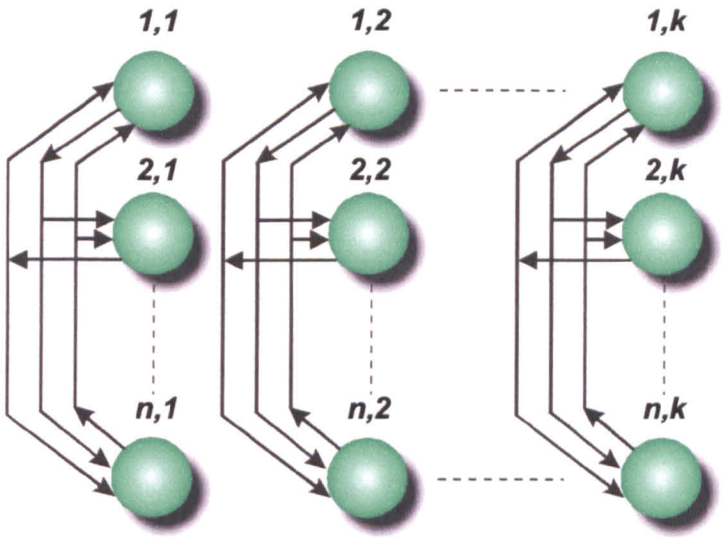


Fig.3.8 CHNN proposed by Cheng et al. where n is the number of grey-levels and k is the number of predefined classes

[Poli and Valli, 1997] describe the segmentation as a small set of two-dimensional layers of neurons, an idea taken from the colouring process of geographical maps (Figure 3.9).

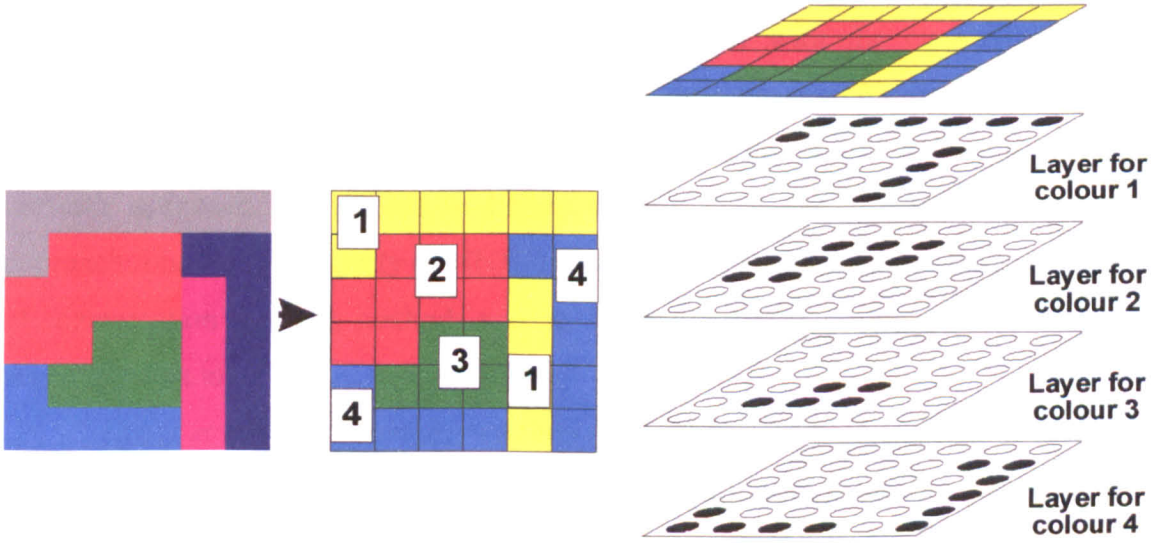


Fig.3.9 Segmentation of a synthetic image (left) using only four colours (center) and the binary representation with a layer of neurons for each colour used (right)

A HNN based on that binary representation for the segmentation of X-ray and MR images was proposed. The semantic energy function includes the sensitivity energy that forces the network to reveal any changes in the image grey level and robustness energy that is aimed to reduce the effects of noise and texture. Moreover, an extension of this approach is proposed for 3D-image segmentation. A very similar approach is described in *[Valli et al., 1998]* for the segmentation of synthetic and CT images.

As a solution to the fact that a single-layer HNN has major limitations, such as settling into a local minimum, a multi-layer Hopfield neural network (MLHNN) for object recognition is presented by *[Young et al., 1997]*. The architecture converges to a local minimum, but that minimum is "often equal or very close to the global minimum". An extended one-layer version of the MLHNN is also presented. However, the computational overhead induced by these proposed architectures is not presented. Furthermore, the speed of the convergence process, which is directly dependant on the number and complexity of computations is not compared with that of a single-layer HNN but with different types of updating procedures. Therefore, more investigations should be done to prove the robustness of MLHNN.

A multi-modal image segmentation method was proposed by *[Rout et al., 1998]*. The method generates a threshold surface by interpolating the image grey levels at points where the gradient is high (possible edges). A modified HNN was used for the interpolation process. The performance is compared with the classical method of interpolation with potential surfaces, where the discrete Laplacian operation is computed for every pixel of the image, except the edge pixels. Although, the neural network approach has proved to be up to five times faster than the conventional approach, due to the property of HNN optimisation, a valuable comparison of goodness of segmentation for both methods was not performed. The presence of noise was not investigated at all.

[Koss et al., 1999] investigated the use of HNN for segmentation of CT or MR images of abdominal organs. The first step to segmentation was to pre-process the images by computing second-order statistical texture transforms *[Haralick et al., 1973]*. The results are then input into a HNN. The HNN structure is presented in Figure 3.10.

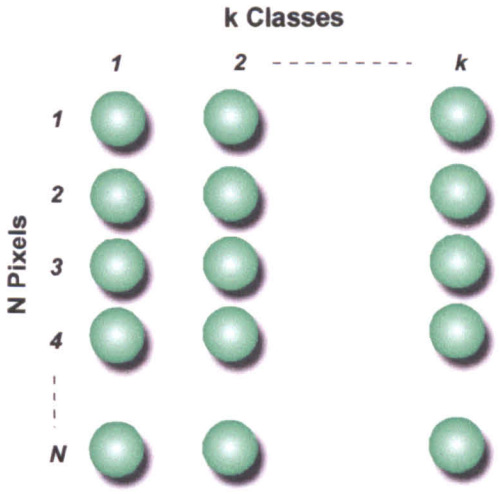


Fig.3.10 HNN structure proposed by Koss et al.; a two-dimensional matrix of neurons: the size of the rows being the number of predefined classes *k*, and the size of the columns being the number of pixels in the image *N*; an active neuron *(i,j)* means that the pixel *i* is assigned to predefined class *j*.

The energy function proposed in this case is:

$$E = A\left[\sum_{i=1}^{N}\left(\sum_{p=1}^{k}\sum_{q\neq p}^{k} v_{pi}v_{qi} + \sum_{i=1}^{N}\left(\sum_{p=1}^{k} v_{pi} - 1\right)^2\right)\right] + \sum_{p=1}^{k}\sum_{i=1}^{N} R_{pi}v_{pi}^2 , \tag{3.9}$$

where R_{pi} is the distance of point *i* from centroid of class *p*. The first two terms combine the constraint conditions of the segmentation (the syntactic energy), i.e. each pixel must be assigned to only one class so that each row must have only one neuron active and the summation of each row must be one in order that a pixel is assigned 100% to a class. The semantic energy is defined by the third term that forces the network to find the minimum of the summation of the distances between the centroids of the predefined classes. In order to decrease the computation overhead and memory requirements, a WTA scheme is applied so that the energy equation (3.9) is transformed like in *[Amartur et al., 1992]*(7):

$$E = \sum_{p=1}^{k}\sum_{i=1}^{N} R_{pi}v_{pi}^2 \tag{3.10}$$

The performance of this method was proved to be above 90% of the pixels correctly classified for most of the organs present in the abdominal cavity. The presence of noise in the original images was not investigated. The authors proposed a series of methods of improvement, such as speeding up the convergence process, reducing the number of neurons *[Cheng et al.1996],* and using a continuous-HNN.

3.2.2.2.2 Self-Organising Maps (SOMs)

[Valli et al, 1998] proposed an architecture based both on anatomical or biological knowledge and the physics of image generation. The overall structure of their system is presented in Figure 3.11. The input image is transformed using image generation physics i.e. enhances information related to signal decay and improves the signal-to-noise ratio of the image. Furthermore, anatomical knowledge is used to generate an initial predefined segmentation of the original images. The resultant images are then fed into the neural classifier that performs the segmentation. The modules that incorporate the anatomical and image generation physics are based on BPNN, while the neural classifier consists of a one-dimensional Kohonen self-organising map used jointly with a unit labelling algorithm *[Wolpert, 1992].*

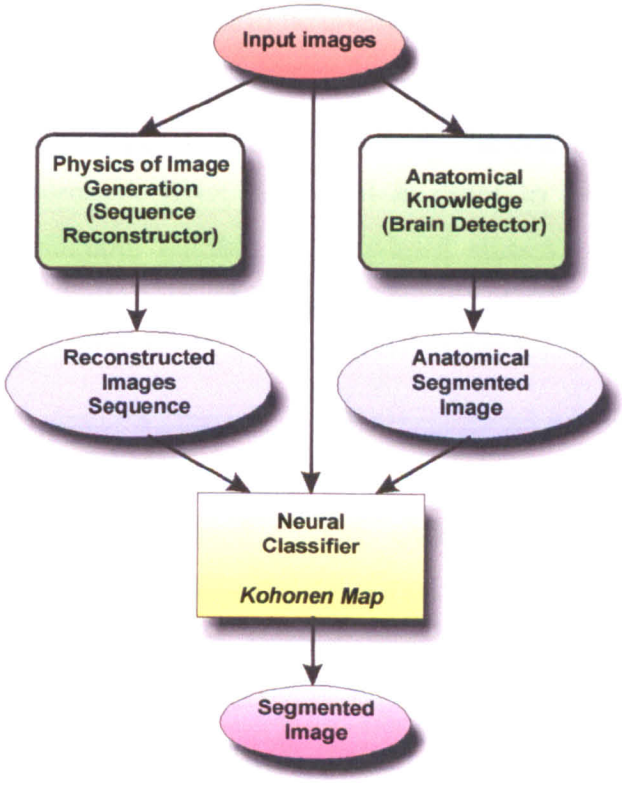


Fig.3.11 Segmentation system based on both anatomical knowledge and physical image generation

A Kohonen self-organising map with a competitive learning algorithm was used by *[Reddick, et al., 1997]* to segment MRIs. This method makes use of T1-weighted, T2-weighted, and proton density-weighted MR images as inputs for the SOM. Thus, there are no spatial constraints involved. A nine-level grey-scale image was the result of the segmentation. After segmentation, a three-layer BPNN was trained to classify each pixel according to the tissue type. In this case, the segmentation was made fully automated, but the classification process still requires *a priori* knowledge, used to train the BPNN. Therefore this method could be classified as a hybrid method between supervised and unsupervised image segmentation techniques. No noise influence was analysed.

A two-step method for segmenting multispectral satellite images was proposed by *[Ambroise, et al., 2000]*. The distribution of the pixels to be classified is analysed using a Probabilistic Self-Organising Map (PSOM) and then Agglomerative Hierarchical Clustering (AHC) is used. Because AHC cannot be used for clustering millions of objects, PSOM provides an initial partition of the image, so that AHC can be applied afterwards. The technique proved to have results that were comparable with classical techniques. Furthermore, the authors claim that different levels of classification are possible using a process that needs to be investigated in the future.

A Kohonen SOM can also be used to speed up the segmentation process. *[Papamarkos et al, 2000]* treated the multi thresholding segmentation techniques as a feature clustering problem. A KNN was used in order to "self organise" the input feature vector to a minimum number of grey-level classes. An extension was also proposed for the segmentation of color images that consists of a Principal Component Analyzer and a KSOM.

3.2.2.2.3 Constraint Satisfaction Neural Networks (CSNN).

[Lin et al., 1992] proposed a new class of neural network for image segmentation – Constraint Satisfaction Neural Networks (CSNN). This method is based on the assumption that an image segmentation problem can be described as a Constrained Satisfaction Problem (CSP). The segmentation problem is redefined as the process of assigning each pixel a label according to certain spatial constraints. A CSNN consists of $n \times n \times m$ neurons, where $n \times n$ is the image size and m is the number of classes the objects needs to be segmented into (called 'labels' – see Figure 3.12 a). Each

neuron is connected to all its neighbours (8 connections, Figure 3.12 b). Those connections are the spatial constraints on the segment label of each pixel. Thus, a CSNN comprises a set of objects, a set of labels (classes) and spatial constraints describing the relationship between various objects according to neighbour relations. They are updated so that a neuron will excite other neurons belonging to the same class and inhibit the others. A winner-take-all scheme is chosen in order to deal with the crossover between segments. As in the standard learning algorithms, there are two phases: learning phase and categorization phase. Each neuron receives feedback from its own output and excitatory/inhibitory signals from its neighbours. This method requires that the number of classes (labels) be known a priori. The proposed method also requires an initial assignment of label probabilities to every pixel using algorithms such as K-means, ISODATA, fuzzy c-means or Kohonen's self-organising map (SOM).

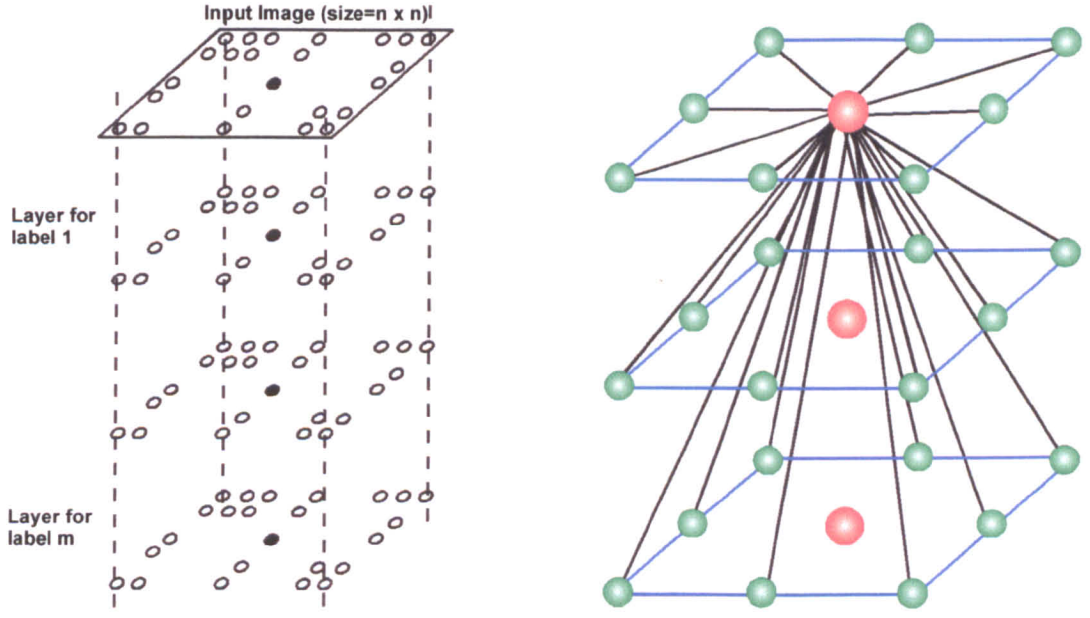


Fig.3.12 a) CSNN Architecture (left); b) Each neuron is connected to all its neighbours (right). The weights of these connections represent the constraints. The network will converge when all the constraints are satisfied with a chosen probability.

Certain improvements to the CSNN method were done by *[Kurugollu et al, 1996]* using explicit edge constraints and *[Kurugollu and Sankur, 1998]* using pyramidal constraints, speeding up the process of convergence near the edges of the images. Another architecture based on a CSNN was proposed by *[Kurugollu and*

Sankur, 1999] – Multi-Scan constraint satisfaction neural network (MS-CSNN). The major advantage of this approach is that the appropriate number of segments is automatically determined by using a modified *[Zhang-Modestino, 1990]* cluster validity index.

Due to the large number of neurons ($n \times n \times m$), this method requires computations that usually are not suitable for real-time applications. It was applied successfully to CT (computed tomography) images and MRIs (*[Lin et al., 1992], [Kurugollu et al, 1996], [Kurugollu and Sankur, 1998]*). Also, there is no literature about the behaviour of this CSNN in the presence of noise.

3.2.2.2.4 A locally excitatory globally inhibitory oscillator network (LEGION)

A locally excitatory globally inhibitory oscillator network (LEGION) was proposed by *[Terman and Wang, 1995]* as a possible computational method for image analysis and segmentation. The idea under LEGION lies in the biological world and is based on the temporal correlation theory. Neurons from the visual cortex respond only to stimuli from a particular part of the visual scene. Moreover, synchronised behaviour had been observed between spatially separated neurons so that cortical neurons corresponding to a distinct homogenous area are oscillating in phase, whereas neurons corresponding to different areas are out of phase. LEGION is able to group similar features in an image by achieving fast synchrony with local excitation. Furthermore, dissimilar features are separated by desynchrony with global excitation/inhibition.

[Chakravarthy et al., 1995] proposed a theoretical network of oscillating neurons for image segmentation purposes - an ANN of complex-valued neurons that exhibit stimulus-specific oscillations. In their model, the synchronization of neural oscillations is produced by both co-operation among neurons via excitatory/inhibitory couplings and by hebbian-like synaptic modification. Therefore, the network exhibits coherent oscillations according to the input images, segmentation occurring due to local uniformities in image intensities. Special local features such as texture, orientation, were suggested to be used to achieve synchronisation between neurons.

[Shareef et al., 1999] used a single layer LEGION-based neural network for segmentation of medical images. A one-to-one correspondence between image pixels and neurons is used. Due to the dynamics of LEGION architecture, pixels with similar features will lead to oscillatory behaviour from their corresponding neurons and their

phases will work in synchronization. Their adaptive scheme for grouping similar features (intensity contrast of pixels) has proven to work better on CT images rather than MR images. The speed achieved was as much as ten times better than classical segmentation techniques, i.e. statistical methods, global thresholding, active contours-snakes, etc. This architecture can be made more flexible, in the sense that a second layer can be added to process the result from the first layer, thus improving the segmentation results.

3.2.2.2.5 Pulse-coupled neural networks (PCNNs)

Recently, pulse-coupled neural networks (PCNNs) have been used for image processing, analysis, including image segmentation *[Ranganath and Kuntimad, 1999], [Kuntimad and Ranganath, 1999].* PCNN is a single layered two-dimensional neural network. It consists of laterally connected pulse-coupled neurons that are a modification of an Eckhorn's neuron model *[Eckhorn et al., 1990].*

The architectures proposed, establish a one-to-one correspondence between image pixels and pulse-coupled neurons. In *[Kuntimad and Ranganath, 1999]* conditions that lead to the "perfect segmentation of a two-object image" (object and background) are derived. None of the other image segmentation techniques described in the present study propose conditions that guarantee a "perfect segmentation".

[Liu et al., 1999] proposed an architecture based on both LEGION and PCNN used to segment range images. In their proposed architecture called Relaxation Oscillator Network, each oscillator has excitatory lateral connections to the oscillators in its local neighbourhood as well as a connection with a global inhibitor. Depth, surface normal and mean and Gaussian curvature values are estimated from pixel location and are used as feature vectors associated with each oscillator. The lateral connection between oscillators is based on a measure of similarity between their feature vectors. There is no need for a priori information and the network is guaranteed to converge rapidly under general conditions. It is suggested that a major application of the proposed architecture is the machine perception, where the range images play a very important role.

The novelty of PCNN and LEGION and their application to image processing/segmentation leads to more research and analysis in order to prove their robustness and applicability.

Neural-network based techniques were proven as successful for the segmentation process. However, in most of the cases, the design and architecture used is context dependant and the implementation of algorithms for different applications can be proven to be difficult. General neural network performances evaluation metrics are not present in the literature. An application dependant knowledge database obtained from human expert experience is applied in a vast majority of cases as an aid for the segmentation process.

All the techniques presented so far, either classical or NN-based are based on hard or crisp variables input. When dealing with an image, a better understanding might be obtained by using fuzzy sets. It is difficult to delineate clear boundaries among human concepts because most of them are vague. Fuzzy sets provide an effective mean to combine the real world with human logical knowledge. Thus, fuzzy-logic based image segmentation techniques will be presented.

3.2.3 Fuzzy approaches to segmentation

Consider the problem of object recognition from an image. Any hard thresholding of the image made for the extraction of the object from the background will introduce uncertainty of decision to subsequent stages. This fact will affect the results of the analysis and in the end the correct recognition of that object. Therefore, it is convenient not to commit to a hard decision (thresholding, edge detection), but rather having the segments or clusters to be fuzzy subsets. While classical or crisp approaches stand for bivalent logic, fuzzy sets pay regards to stages between clear-cut conditions.

A general schemata of a fuzzy image segmentation process is depicted in Figure 3.13. The input to the system is the original image that needs to be partitioned into objects or clusters. Using an expert knowledge base, features are extracted from the image, i.e. homogeneity of a zone, the transition between two zones, etc. These features are considered as inputs to the fuzzy inference system. They are first fuzzified, a process that transforms the inputs into membership values belonging to some defined linguistic terms. This stage depends heavily on the definition of the membership functions. Reasoning stage is based on the definition in natural language of the application-specific segmentation. For instance, for edge detection, natural language can be used with the following formulation: "an edge corresponds to a high transition between two

compact areas". This natural language definition is translated into fuzzy rules that are evaluated and a decision is taken for each area or pixel in question as to what membership degree it belongs to a cluster or an edge. The last stage of the fuzzy inference system is the defuzzification process in which the degrees of membership to an object or clusters or edges are translated into precise or crisp values. As it can be seen from Figure 3.13, the knowledge base provides at every stage information regarding the system's parameters and it is context or application dependant.

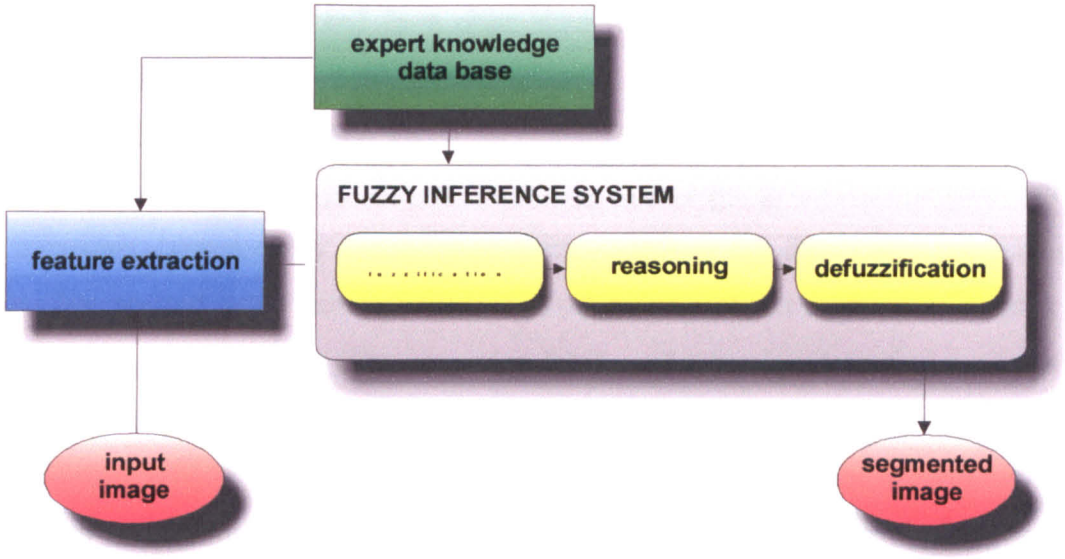


Fig. 3.13 Generic fuzzy approach to image segmentation

There are several fuzzy methods proposed for image segmentation, either for iterative pixel classification or edge detection. Furthermore, fuzzy techniques have been used as more high-level image processing, such as feature extraction and object recognition.

[Cho and Cho, 1994] used a fuzzy technique as an alternative to adaptive thresholding. Contrast and local brightness are the input measures to a fuzzy inference system and the output is the local threshold. A 3x3 mask is applied to every pixel in the image. In order to decide the degree of threshold values according to image context, an output membership function is generated subject to the standard deviation of the input image.

As an edge detection based segmentation method, *[Jendrysik et al, 1997]*, proposed a technique for the extraction of kidney boundaries from medical images. The fuzzy algorithm is used to determine whether a pixel belongs to an edge (contour) or to

the object or background. Fuzzy IF-THEN rules determine the contour membership grade of every pixels using neighbourhood information. For every pixel, the information provided by its eight neighbouring pixels is used to determine its membership grade. Four masks were used so that only two neighbouring pixels are taken into account at one time. A fuzzy inference system with 9 rules was designed and implemented with promising results. Furthermore, a post-segmentation method, Contour sorting, based on fuzzy logic, is introduced in order to improve contours in matters of regularity or thickness *[Waidelich et al., 2000]*. The method is able to distinguish between the contours of a given object and from other detected contours with the aid of two fuzzy variables and six fuzzy if-then rules.

A similar approach was taken by *[Jaulent et al., 1997]*. Renal angiograms are segmented using a three-stage system. First stage is an edge detection operator based on two measures: homogeneity and heterogeneity - small or big differences between grey-levels. The second stage corresponds to the detection of the regions of interest that encompass the renal arteries. The last stage implements an edge detection operator with a fuzzy clustering based on the Fuzzy-C-Mean (FCM) algorithm. In the FCM method *[Bezdek et al., 1997a], [Bezdek et al., 1997b]*, each class is represented by its centre and the goal is to minimise a cost function based on the summation of the quadratic errors inside a class between each element of the class and the centre. In the FCM method, every pixel belongs to all classes with different degrees of the membership function, as opposed to the Hard C-Means Algorithm (HCM), another segmentation technique, where each pixel belongs only to one class. The algorithm of FCM can be summarised as follows:

FCM algorithm

1) randomly choose the initial class centers z_i , $1 \leq i \leq k$, where k is the total number of classes; compute the histogram of the image (pairs of values h_x and grey level g_x); compute the square Euclidian distance measure between the grey levels g_x and the class center z_i for all classes :

$$d_{x,i}^2 = \|g_x - z_i\|, \text{for } 1 \leq x \leq n, 1 \leq \mathrm{i} \leq \mathrm{k} \tag{3.11}$$

where n is the total number of grey-levels present in the image

2) compute the membership matrix U and update the class centers z_i:

$$\text{a.} \quad u_{x,i} = \frac{\left[\frac{1}{d_{x,i}^2}\right]^{\frac{1}{m-1}}}{\sum_{i=1}^{k}\left[\frac{1}{d_{x,i}^2}\right]^{\frac{1}{m-1}}} \tag{3.12}$$

$$\text{b.} \quad z_i = \frac{1}{\sum_{x=1}^{n}(u_{x,i})^m h_x} \sum_{x=1}^{n}(u_{x,i})^m h_x g_x \tag{3.13}$$

where m is called the exponential weight (the fuzzification parameter) and it will increase the noise effect when computing the class centres; the higher the value of m, the more the sensitivity to noise will be.

3) check convergence and stop, otherwise go to step 2)

[Hall et al., 1992] used a FCM algorithm for the segmentation of magnetic resonance images of the brain, compared with a dynamic multilayered perceptron trained with a cascade correlation algorithm. Their approach has a high computational overhead making it impossible for use in real-time applications involving MRIs. *[Cannon et al.,1986]* proposed a variant of the same FCM algorithm that reduces computational overhead. A Modified version of FCM (MFCM) is used by *[Betanzos et al., 2000]* for the segmentation of patients with burns. It consists of two phases: one for the determination of the number of classes or clusters and the actual fuzzy phase when the FCM algorithm is applied. This algorithm's performance was compared with the Fuzzy K-Nearest Neighbour (FKNN) algorithm performance. After the typical elements for all classes or clusters of the image are labelled, one possibility to assign a new pixel to a variable is given by the k-nearest neighbour rule, where k is the predefined number of neighbours taken into account for the decision. For each element that needs to be classified, FKNN determines the value of the correspondence function for each one of the nearest k samples, known as "neighbours". The equation used to compute a pixel's factors of correspondence to the clusters being considered is:

$$u_x(x) = \frac{\sum_{j=1}^{k} u_{ij} \left(\frac{1}{\left\| x - x_j \right\|^{\frac{2}{m-1}}} \right)}{\sum_{j=1}^{k} \left(\frac{1}{\left\| x - x_j \right\|^{\frac{2}{m-1}}} \right)} \tag{3.14}$$

where u_{ij} represents the factor of correspondence of the j^{th} sample to the i^{th} class; x_j represents one of the k samples nearest to the current pixel x; m is a parameter of the algorithm, similar with m for the FCM algorithm.

The above approaches and algorithms do not include other information in the decision process other than of the grey-values and histogram information. *[Tolias et al., 1997]* imposed spatial constraints in the image segmentation problem. A fuzzy inference system with 3 input measures and 11 rules that interacts with the clustering results obtained by FCM was designed and implemented. It was based on the behaviour of Gibbs Random Field model (GRF). The results showed lower computational overhead compared with stochastic field modelling segmentation algorithms.

Implementation of FCM can also be applied at lower level image processing, as an initial segmentation for automatic segmentation of brain tumors in MRIs *[Fletcher-Heath et al., 2001]*. The MR feature images used for the segmentation consists of three weighted images or slices (T1, T2 and proton density PD). A control knowledge-based system integrates domain knowledge as a high-level tumor segmentation. A similar approach was taken by *[Hirano and Hata, 2001]* in an attempt to use a fuzzy expert system for the decomposition of a fractured foot CT image into fragments and normal bones. Two types of knowledge are integrated: knowledge of distance to extract fragments and knowledge of intensity distribution to decompose adjoining normal bones on the articular cartilage.

Detection of outlines from an image based on the principle of fuzzy logic is proposed by *[Engel et al., 1997]*. The guiding principle for the approach was that that the human visual system is the best detector of outlines. Therefore the authors proposed an imitation to this system. An outline is interpreted as a discontinuity of the intensity,

on a set of "spots". Fuzzy logic is then used to characterize the intensity from a spot and the membership value of the spot to an outline. Tests performed on more than 100 images proved the applicability of this method to a first stage of machine vision and scene interpretation.

The use of fuzzy sets in edge detection, noise elimination and object smoothing are presented by *[Keyserlingk and Pohl, 1997], [Berks and Keyserlingk, 1998], [Berks and Keyserlingk, 2000]* for three-dimensional MRI and CT data sets. A pixel is characterised by its grey value and its neighbourhood. Linguistic variables for the fuzzy system were defined for one pixel and its three-dimensional neighbourhood (a 3 by 3 by 3 mask was used for each pixel). The advantage of the presented approach consists of its great flexibility and the easy rule-based conveyance from real situation in the image.

An edge detector operator based on fuzzy liguistic rules is presented by *[Bombardier et al., 2000]* as a mean to introduce "high-level information" in low level image processing such as edge detection in order to accommodate a specific context of the implementation. The image quality information is obtained either from the histogram information or from an expert assessment and it is used to reiterate and modify the membership functions of the edge operator.

The nature of fuzzy logic and its representation with the real world human knowledge requires the use of many parameters. To determine the optimal values for parameters, both in fuzzy and neural-network based image segmentation techniques, experiments are employed, an obvious time consuming and cost generator process. Thus, genetic algorithms have been applied as search methods for optimal parameters values.

3.2.4 Genetic algorithms

Genetic algorithms (GAs) are search methods that have also been employed in the segmentation processes, mostly as a way to speed up the process or to calibrate its parameters. GAs are mostly used in order to solve an optimisation problem. They need to maximize a fitness function depending on the nature of the problem. Each solution is coded as a binary string called chromosome, grouped in populations. The initial randomly chosen population evolves to a next generation using selection, crossover and

mutation genetic operations. The evolution process is iterated until an optimal solution is obtained or a predefined number of generations is reached.

[Yin, 1999] proposed a genetic algorithm based technique for speeding up the multilevel thresholding process. Because GAs can perform parallel search, it is feasible to apply them to accelerate the optimal thresholding methods. Yin used a GA in conjunction with the optimal bilevel thresholding technique proposed by Otsu (*op cit).* The results proved that the technique can accelerate the optimal thresholding methods and the quality of the thresholded images is better than those of property-based multilevel thresholding methods.

Edge detection can also be defined as an optimisation problem; therefore GAs can be applied to solve it. *[Gundmusson et al., 1998]* proposed an algorithm that detects well-localized unfragmented, thin edges in medical images based on optimisation of edge configuration using a genetic algorithm. The segmentation process of images composed of different kinds of textures is viewed as an optimisation problem also by *[Yoshimura and Oe, 1999].* The automatic segmentation method can decide the optimum number of segmentation areas in a texture image composed of similar texture fields.

3.3 Conclusions and recommendations

There are many image processing techniques present in the literature. Only a selective survey was presented in this chapter. Most of the methods are used for a specific problem and are context-dependant on their application implementation. Among the image processing techniques, a very important role is played by the high-level segmentation process. Consequent image processing and analysis is highly dependant on its results (such as in an image-based inspection system) and therefore an increased attention was given to defining novel and better approaches. There are several definitions of the segmentation process and most authors create their own definition according to their solution or their context-dependant application.

Most segmentation methods are relying on the grey-level distribution and values of the pixels within the image that needs to be segmented. Classical techniques usually involve some sort of statistical computations that would eventually lead to segmentation. When dealing with blurred or fuzzy images, like in the case of X-ray

images where edges are not sharp, classical techniques cannot be applied very successfully. Morphological operators for X-ray image segmentation were used by *[Graves, 2000 – personal communication]*. However, due to the fragmentation of the image, this method is not suitable for the inspection of foreign bodies (will lead to increased number of computations when fragmentation occurs). Neural-network based techniques are used successfully for the segmentation process. The results presented in the literature promise these methods a bright future, though much more work and research needs to be done. In the vast majority of the papers reviewed here, the results of the segmentation process lack general recognised metrics or a general evaluation framework. Therefore, comparison between methods is still a subjective matter of opinion. Moreover, one of the best advantages of neural networks – their graceful degradation in the presence of noise- has not been thoroughly investigated.

Many methods presented here require sample data and *a priori* knowledge such as features extracted from training images or k - the number of desired classes or regions to be segmented.

No general applicable framework or generic method has been found for image segmentation. Although the possibility of such a method is suggested by some authors *[Ranganath and Kuntimad, 1999], [Kuntimad and Ranganath, 1999]*, further research is needed to prove its robustness and reliability. The next chapter describes a specific context of image processing from which generic ideas are developed.

"A year spent in artificial intelligence is enough to make one believe in God."

Alan J. Perlis

Epigrams of Programming

C H A P T E R 4

A SYSTEM FOR IMAGE PROCESSING

4.1 Introduction

The specific problem to be investigated in this work is the one of the detection of foreign bodies within chicken breast meat (especially the detection of the presence of bones in the final meat product). A general schemata of the prototype IMS Bonescan system used in the experiments is presented in Figure 4.1. Its main applicability is in the detection of foreign bodies within food products.

Such a system comprises an X-ray source, an image intensifier, a TV-camera and a conveyor-belt. The product is passed through the beam emitted by the x-ray source. Below the conveyor belt the image intensifier gathers the remaining x-rays and converts them into a visible image. A TV-camera is used to record that image on-line and to produce a video signal. The video signal can be fed into a video monitor for visual detection or with the aid of a video-capture card fed into a computer for

automatic post-processing and detection. More details about the processes involved in the running of the detection system are presented in Appendix A.

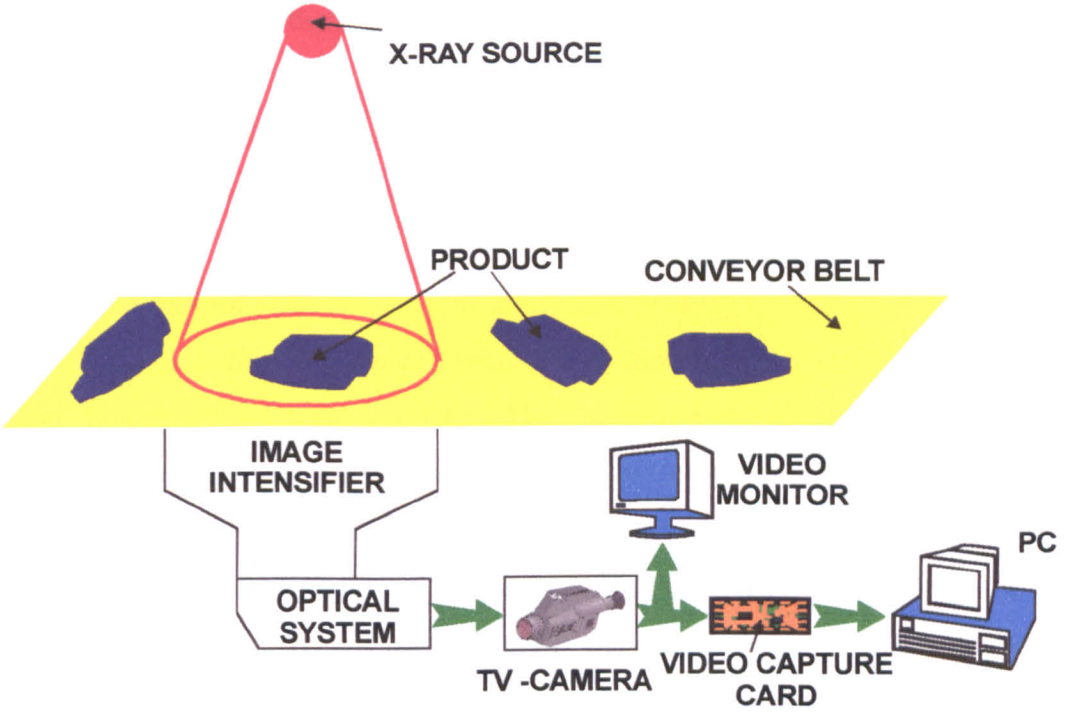


Fig.4.1 System for foreign body detection in meat for the food industry

The proposed image processing system assumes three main stages, as depicted in Figure 4.2.

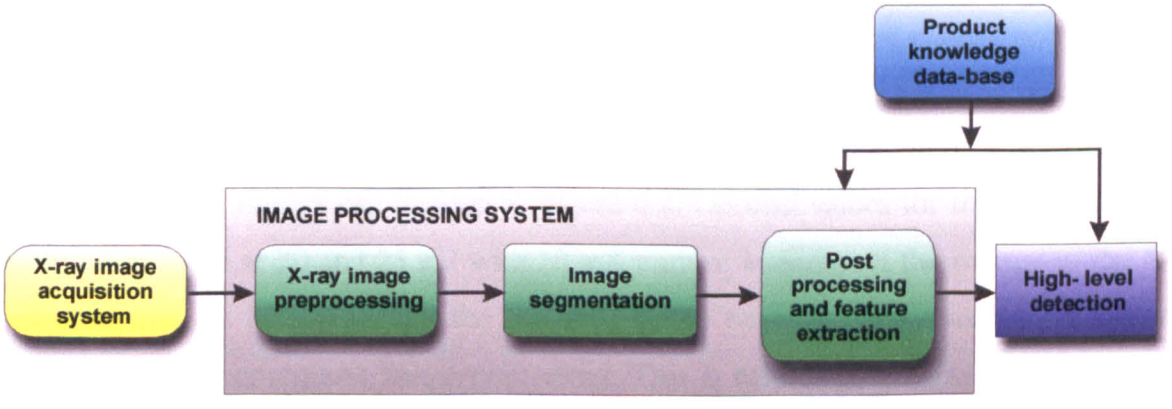


Fig.4.2 The image processing system used

The input of the Image Processing system is the X-ray image acquisition system that provides the X-ray image of the inspected product. The image is then pre-processed

so that its grey-level characteristics are enhanced and possible noise removed. Segmentation is the logical following step, in which the image is divided into meaningful objects. From those objects features are extracted and a decision whether a foreign-body is present in the product is taken.

In the following paragraphs, each stage of the Image Processing system is presented in more detail along with the proposed associated solution and algorithms.

4.2 Image acquisition

A black and white TV camera was used to capture X-ray images from the image intensifier unit and fed into a computer. Thus, images with 256 level of grey (768 by 576 pixels) were captured. Since pieces of meat can often overlap, creating a zone of high meat density possible foreign bodies present in that region can be masked. Consequently, in zones of high meat density, possible foreign bodies present can absorb the same quantity of radiation and therefore become merged with the background. Under these circumstances, it was decided that a dual-band image needs to be taken from each inspected product. A multiple band image consists of two or more views of the same area or product, the views being taken under different conditions or using different wavelengths of light or signals. In the present case, the dual-band image consists of two views taken with different X-ray intensities. A low-intensity image and a high-intensity image were taken. The idea behind it is that if a bone is masked in the low-energy image, in a high-energy image it is very probable to be visible due to unequal X-ray absorption coefficients of bones and meat.

X-ray images were taken using different energy values for the radiation. The X-ray source was powered with pairs of 10, 15,18, 20, 25 and 30 kV with 10 mA intensity and the resulting images can be seen in Figure 4.3. Optimal values for the X-ray energy were found to be at 20 and 25 kV with 10 mA current intensity. If the voltage of the X-ray source is set below 20 kV, the resultant image becomes very dark and therefore unusable in the consequent image processing. This is because the X-ray radiation does not possess enough energy to fully penetrate the inspected product and therefore the X-ray detector is unable to retrieve what is left from the original incident radiation. In contrast, when the voltage of the X-ray source is set above 25 kV, a very high percentage of the original X-ray radiation can reach the X-ray detectors; therefore, the

change in the original X-ray radiation energy is minimal and the resultant image is very bright and not usable.

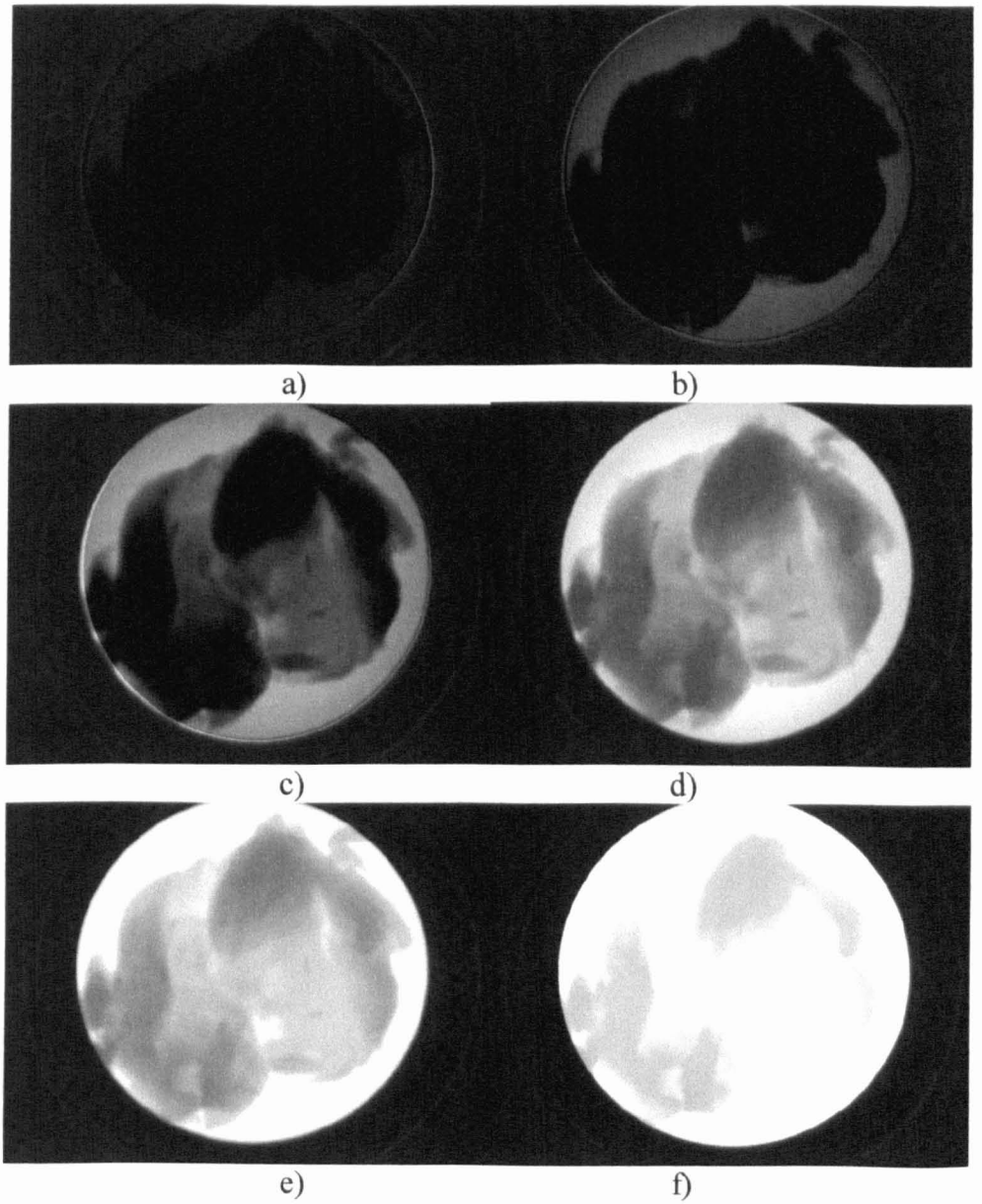


Fig.4.3 Different X-ray energy images

a) 10 kV, 10 mA; b) 15 kV, 10mA; c) 18 kV, 10mA; d) 20 kV, 10mA; e) 25 kV, 10mA; f) 30 kV, 10mA;

Thus, the low-energy view of the product and the high-energy view of the inspected product were taken when powering the X-ray source at 20Kv and 25 kV respectively (at a current intensity of 10 mA).

There is a perfect correspondence between spatiality of the two views, in that that each pixel of the low energy image is located at the same coordinates in the high-

energy view. Therefore, there is no need for registration of images and this is achieved by having the conveyor belt stop for 0.5 seconds, while the dual band images are acquired. There is no movement of the product between the acquisition of the low and high energy images. This is a main aspect of further image analysis, since spatiality and the location of the possible foreign-bodies are of a crucial importance.

4.3 Image pre-processing and calibration

In the X-ray images taken, the raw information is in most of the cases presented in a way unsuitable for the human eye or for further image analysis techniques (poor contrast, presence of noise, etc.).

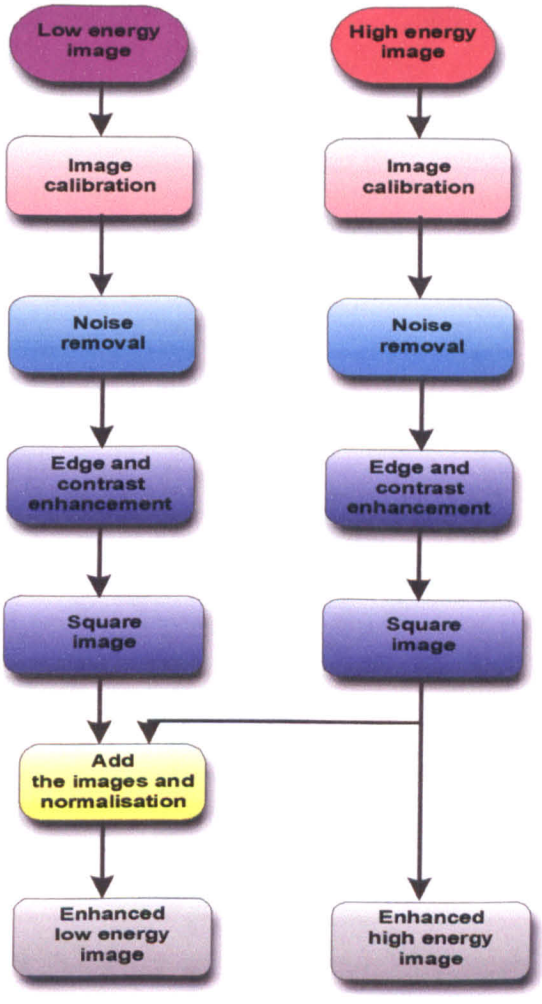


Fig.4.4 Image pre-processing

Therefore, methods for "improving" or enhancing the images were developed. The contrast of an image is given by the distribution of its grey levels or its pixel values.

If this distribution is concentrated near a certain level, then the contrast is obviously low. On the other hand, when a wide range of grey levels are presented in the image, then the contrast is high. It is the main objective of image pre processing to improve images to appear suitable for the human eye. Only after pre-processing, further automated computer based image analysis algorithms can be employed, since all are based on the knowledge gathered through human visual senses. A general flowchart of the pre-processing stages is shown in Figure 4.4.

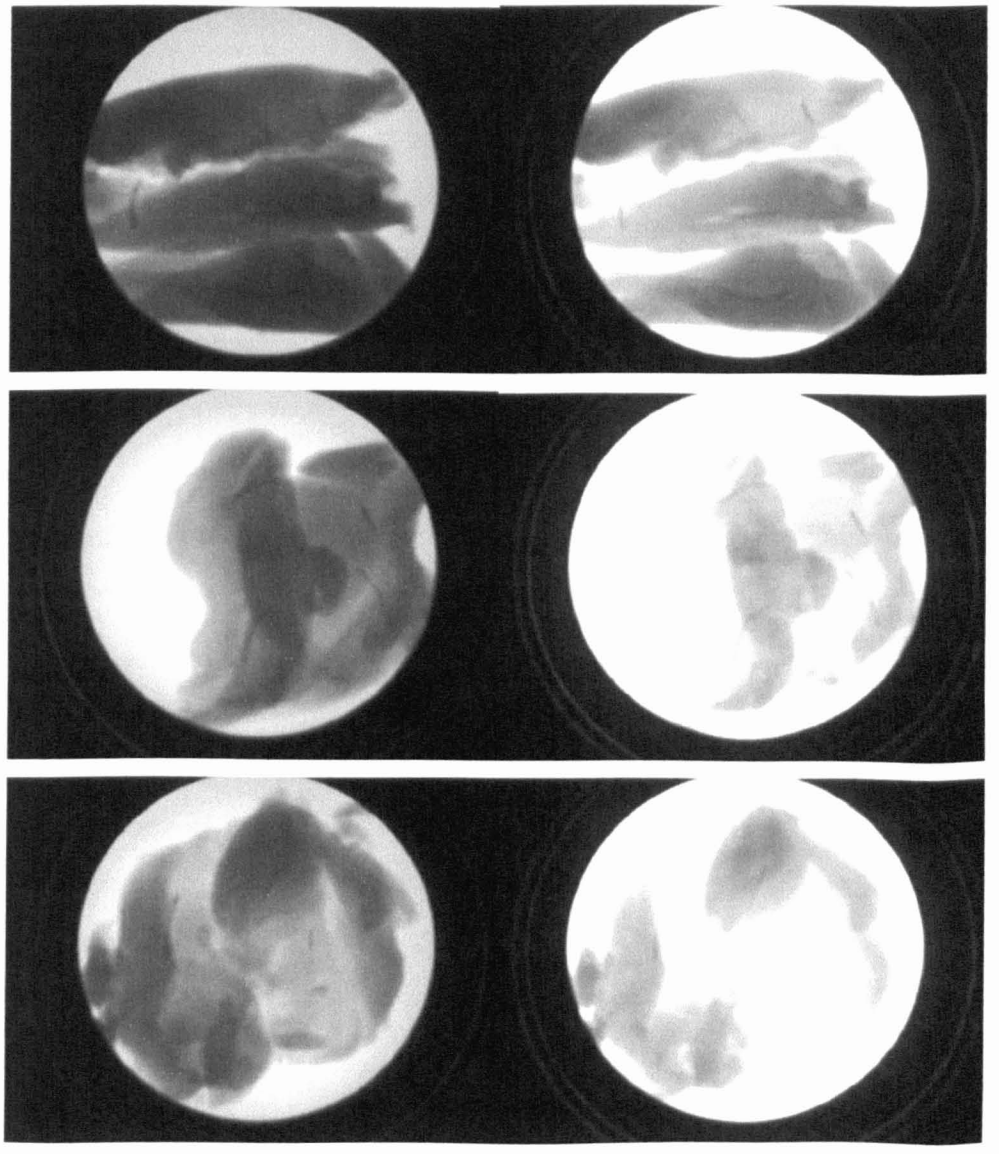


Fig.4.5 Examples of dual-band X-ray images of chicken breast meat containing different number of bones (low energy image on the left; high-energy image on the right)

The images are first calibrated and the possible noise is removed. Then edge and contrast enhancement techniques are employed in order to improve the quality of both low- and high-energy X-ray images. The images are squared, normalised and added together; therefore a composite image that incorporates information from both images is formed. The latter image and the high-energy images are now the components of the single dual-band X-ray image that will be analysed further.

Sample dual-images are depicted in Figure 4.5.

4.3.1 Noise removal

Transmission over cables of video signals is often affected by the presence of electromagnetic fields, bad shielding of cables, etc. Consequently, noise can corrupt the resultant images. The noise pattern consists of strong, spike like components that can affect randomly the grey-value of one or more pixels, called impulse or salt-and-pepper noise. Methods used for noise removal are usually based upon spatial-domain techniques. In the spatial domain, an image-processing function can be expressed as follows:

$$g(x,y) = Transf[(f(x,y)] \tag{4.1}$$

where *f(x,y)* is the input image, *g(x,y)* is the processed image, and *Transf* is a spatial operator on *f* defined over some neighbourhood of *(x,y)*. Usually, a rectangular mask or window centred in (*x,y*) is moved from pixel to pixel . The operator is then applied to compute the processed image *g*.

The image can be smoothed by averaging the grey-level values of the pixels in a predefined neighbourhood. This spatial-domain technique is called neighbourhood averaging and can be successfully applied for the removal of impulse noise:

$$g(x,y) = \frac{1}{t} \sum_{(p,g)\in Neighbourhood} f(p,q), \quad x,y, = \overline{0..N} \tag{4.2}$$

where *t* is the number of points in the neighbourhood. The main disadvantage of this technique is that it blurs edges and other sharp details. Since foreign bodies are considered areas of sharp details, a median filtering was used to remove the impulse

noise. The method consists in replacing the grey-value of each pixel by the median of the grey levels in a neighbourhood (or prespecified window or filtering mask) of that pixel, instead of by the average. In this way, points or pixels with very distinct intensities will be removed and replaced with pixels more like their neighbours. A 3 by 3 pixels window was employed and the median filter was applied for both high and low energy images in two passes. The results can be seen in Figure 4.6.

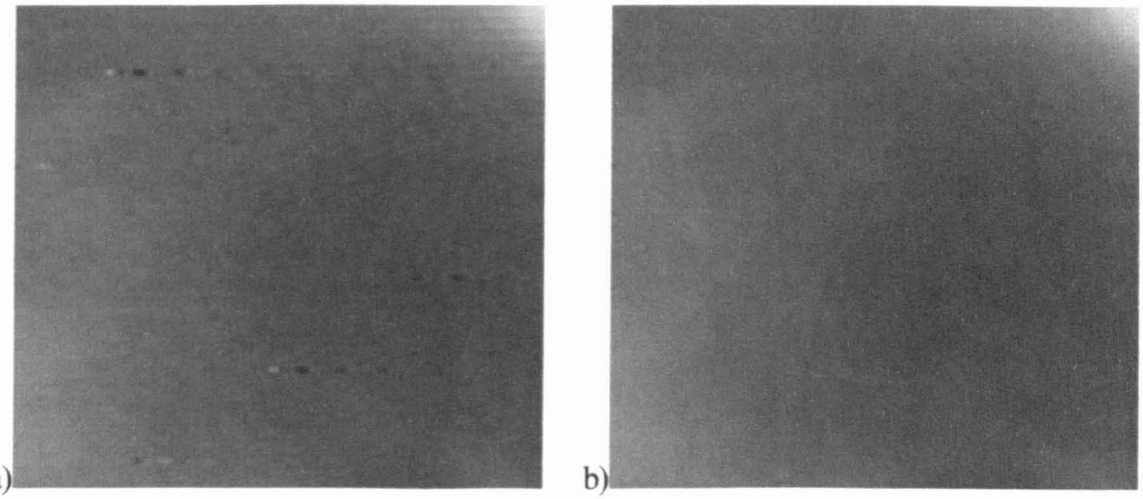


Fig.4.6 Noise removal by using a 3 by 3 pixels Median filter in two passes
a) original image corrupted by impulse noise; b) result after median filtering

4.3.2 Image calibration

Due to the use of an image intensifier, intensified images of a uniform object are generally brighter in the centre than in the periphery due to an unequal brightness gain in different regions of the field of view. This effect is also called ***vignetting*** (see Figure 4.7). Moreover, the performance of the X-ray tube and the fluorescent screen of the image intensifier suffer decay over time. Further analysis of the images may suffer over this effect and therefore, calibration images are taken at the start of each inspection process. An example of image calibration for both low and high energy images are illustrated in Figure 4.8.

By simply substracting the calibration image from the input image, the vignetting effect can be removed and the decay of X-ray acquisition system's performance over time is annulled. There is no need for the registration of the two images and where needed, the result was normalized by dividing the average pixel intensity value of the original image. Also the removal of the bias frame from the actual

image, process called debiasing, is achieved in this way. The substracting process is applied to both low and high-energy images. The results are shown in Figure 4.9.

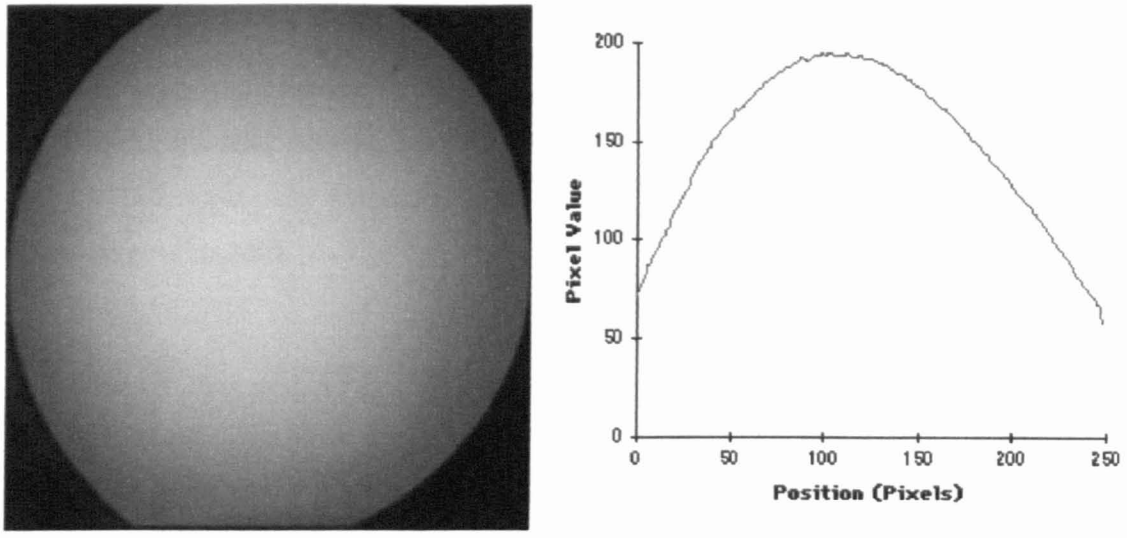


Fig.4.7 Vignetting effect for X-ray image intensifiers

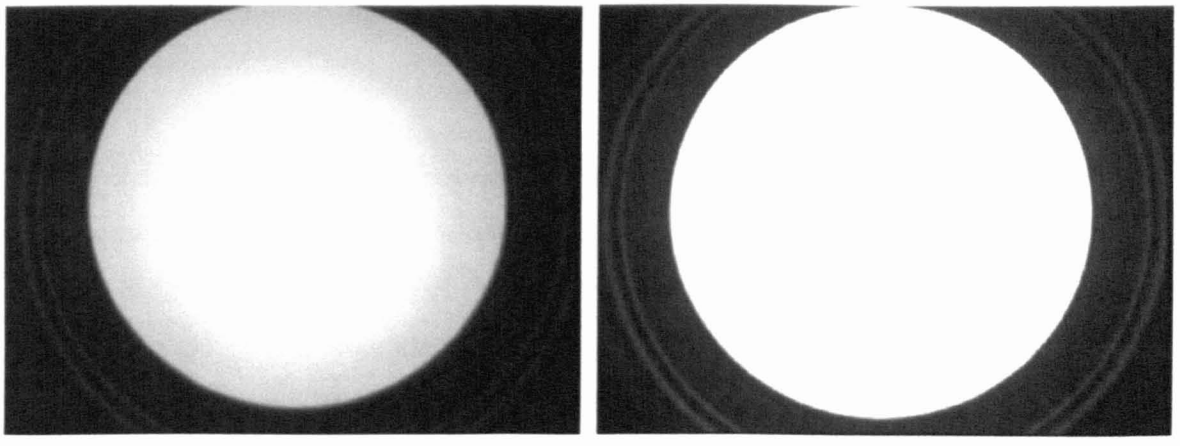


Fig.4.8 Calibration images for low and high energy views

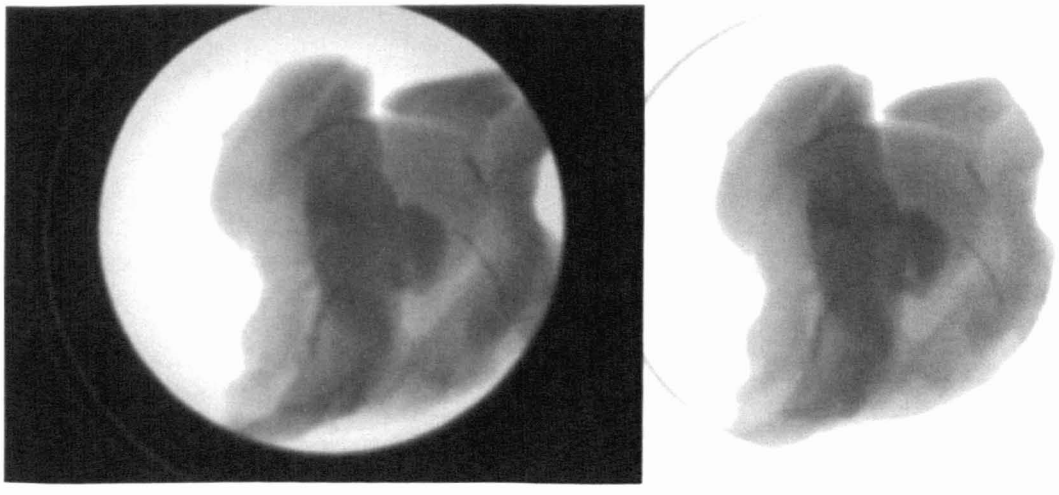


Fig.4.9 Calibration result

4.3.3 Edge enhancement

Since edges are of a main importance where image analysis is concerned because of their properties to emphasise the object within a scene, enhancement algorithms are implemented. In the original images, due to the nature of the X-ray tube and of the image intensifier system, edges are of a low quality and they require sharpening or enhancement. Edge enhancement is realised by using a method from the frequency-domain. The foundation of the frequency-domain techniques is based on the convolution theorem:

$$g(x,y) = h(x,y) * f(x,y) \tag{4.3}$$

where *h(x,y)* is a position-invariant operator. Translating the above relation into the frequency-domain, the following relation can be deduced:

$$G(u,v) = H(u,v)F(u,v) \tag{4.4}$$

where *G,H,* and *F* are the Fourier transforms of *g,h* and *f*. In a typical image-enhancement frequency-domain method, *f(x,y)* is given and after computation of *F(u,*v), the aim is to find *H(u,v)* so that the processed image *g(x,y)* exhibits some highlighted feature of *f(x,y)*. Since edges or other abrupt changes in grey-levels are associated with high-frequency components, image sharpening can be achieved in the frequency domain by a *highpass filtering* process *[Gonzalez and Wintz, 1987]*. Such a filter attenuates the low-frequency components without disturbing high-frequency information in the Fourier transform of the image *[Teuber, 1993]*. Since an ideal highpass filter is not possible, a "Butterworth" filter of order α was used *[Teuber, 1993]*:

$$H(u,v) = \frac{1}{1 + (\sqrt{2} - 1)\left[\dfrac{D_0}{D(u,v)}\right]^{2\alpha}} \tag{4.5}$$

where D_0 is the cut-off distance from the origin of the frequency plane, and *D(u,v)* is given by:

$$D(u,v) = \sqrt{u^2 + v^2} \tag{4.6}$$

Since computational overhead is an issue in real-time applications, the order of the applied filter was 1. The result of applying the "Butterworth" filter is shown in Figure 4.10.

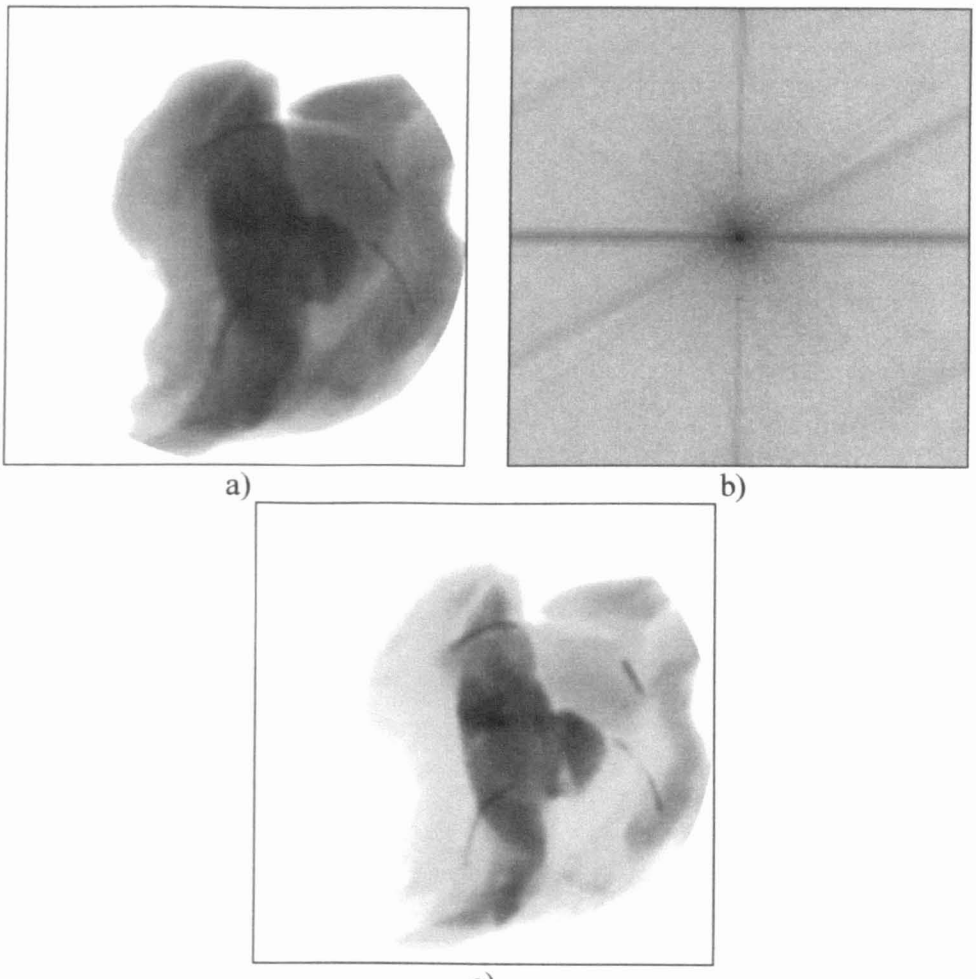


Fig.4.10 Edge enhancement by using a first order "Butterworth" high-pass filter a) original X-ray image; b) Fourier transformation of the image; c) Results of the high-pass filter

4.3.4 Contrast enhancement

Simple arithmetic techniques from the spatial-domain are employed for contrast enhancement of the resultant filtered X-ray images. By multiplying an image by itself (square the image) the intensity values of importance can be enhanced (contrast enhancement). Normalisation can sequentially be performed by dividing the average pixel intensity value of the original image in order to achieve a high contrast image.

This process is applied to both high and low energy images. The resultant images were added to each other pixel value by pixel value and a composite image was obtained whose pixels have similar average intensity to the original images, with much less noise than any one of the given original images. In other words, the resultant images have a much greater *signal-to-noise* ratio. Normalisation must also be performed in order to scale the new pixel's values to a useful range. The result is shown in Figure 4.11.

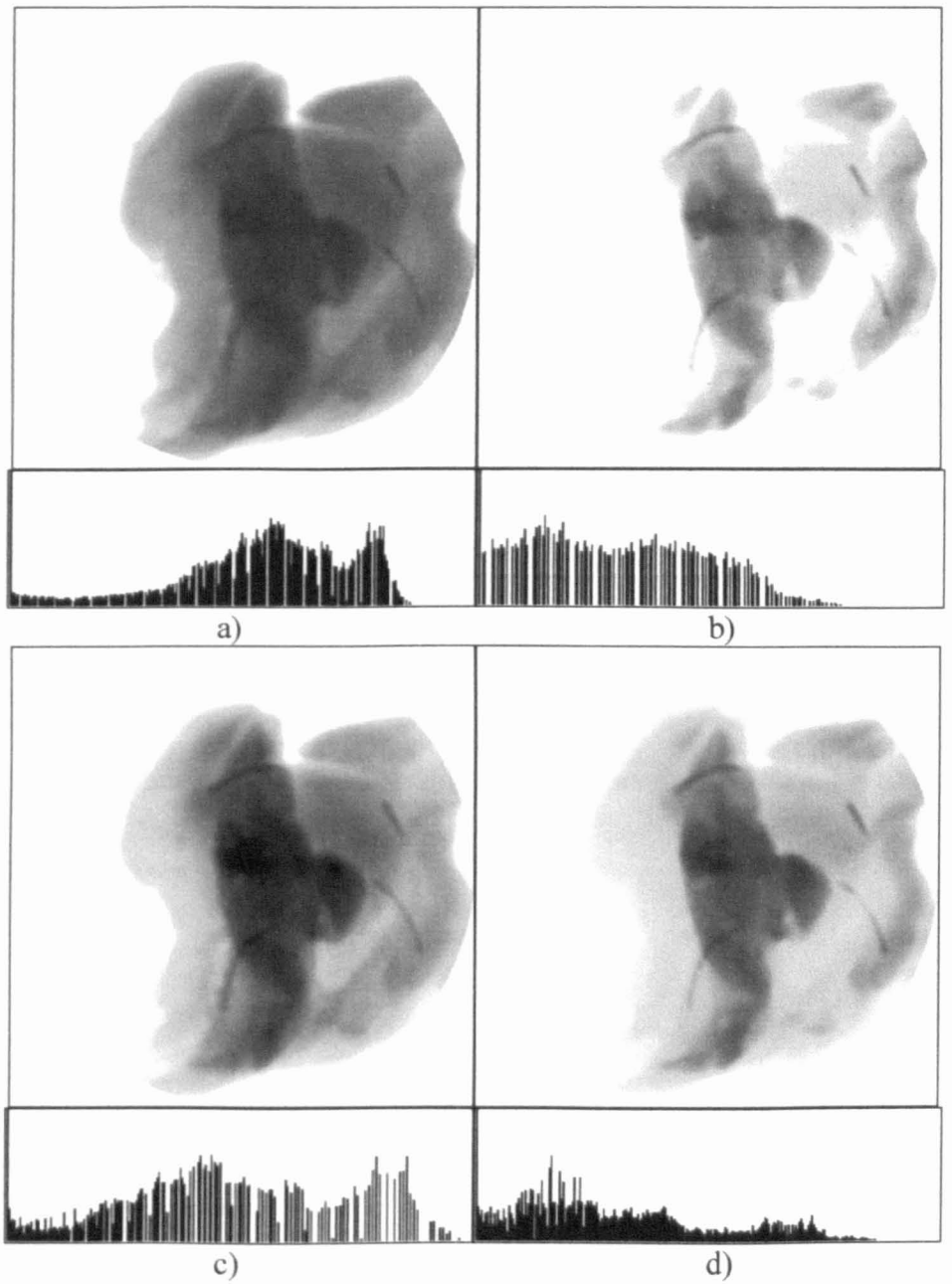


Fig.4.11Arithmetic operations performed on images and their corresponding histograms a) original low energy image; b) original high energy image; c) result of multiplying the original low energy image by itself; d) result of adding between squared high energy image with c)

Thus, two images were the final result of the pre-processing stage: an enhanced low-energy image and an enhanced high-energy image, components of a dual-band image of the inspected product. There is still no need for registration between the images and subsequent image analysis techniques assumes it. Consequently, the next step of the image processing system is the image segmentation process.

4.4 Image segmentation

4.4.1 Classical approaches to X-ray images segmentation

As was explained in the previous chapter, a segmentation algorithm for a meat X-ray image needs to separate foreign objects (such as bones) from the background (the meat itself). One aims in separating not only entire objects from the background, but also separating only parts of objects from the background is also considered a successful technique.

A simple thresholding of a meat X-ray image would provide a useless result for further image analysis techniques. To illustrate this, an Otsu-based (*op cit)* algorithm was implemented. The results are depicted in Figure 4.12. The meat product contains three easily visible bones (Figure 4.12 left). When thresholding the image, bones are merged with other parts of the background (meat) (Figure 4.12 right) and therefore, a correct extraction of important objects is not possible in this way. Therefore, multilevel thresholding techniques need to be employed to solve the segmentation problem.

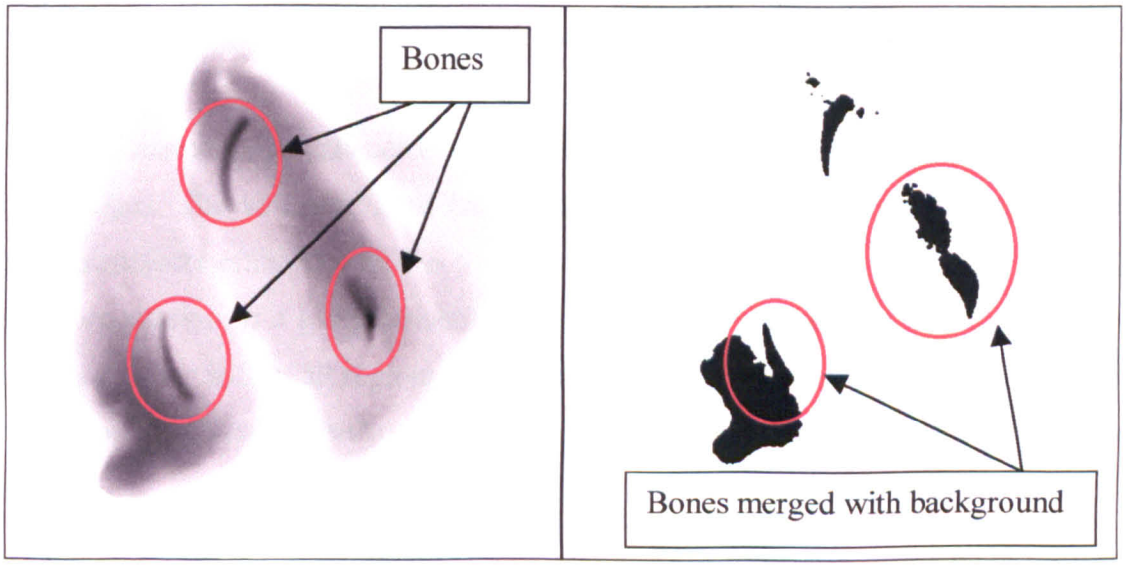


Fig.4.12 Results of Otsu's thresholding method

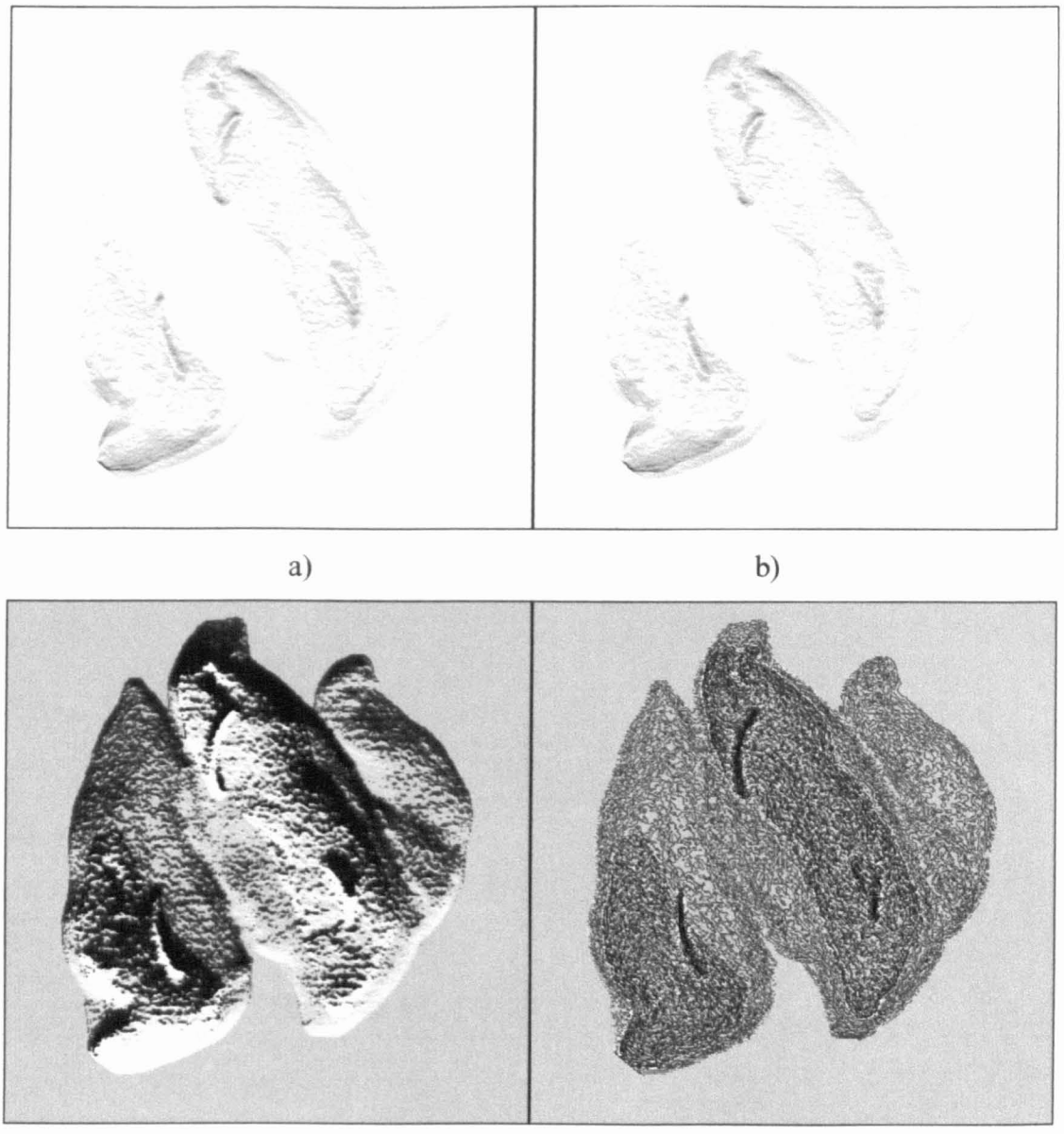


Fig.4.13 Edge detection results using a) Prewitt; b)Sobel; c) Gradient; d)Laplace

Classical edge enhancement and detection techniques were also tested on the X-ray images. The aim here is to have a resultant image that contains contours for the foreign objects embedded in the food product. The idea under lying edge detection is the computation of a local derivative operator. The first derivative of an edge modelled in this manner is 0 in all regions of constant grey level and constant during a grey-level transition. The first derivative of an image is called *gradient* and it is defined as follows:

$$G_{f(x,y)} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \tag{4.7}$$

The computation of the *gradient* of an image consists in the determination of the partial derivatives at every pixel location *(x,y)*. A 3 by 3 mask (or convolution kernel) was used. Thus, applying gradient edge detection is similar to the following convolution operation:

$$g(x,y) = f(x,y) * Gradient \tag{4.8}$$

where Gradient kernel or mask is presented in Table 4.1.

The *Laplacian* edge enhancement technique produces sharper edge definition than most other techniques. Its main property is that it can highlight edges in all directions. The Laplacian of an image *f(x,y)* can be determined as follows:

$$L_{f(x,y)} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{4.9}$$

where

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - 2f(x) + f(x-1)$$
$$\frac{\partial^2 f}{\partial y^2} = f(y+1) - 2f(y) + f(y+1) \tag{4.10}$$

An approximation of the *Laplacian* can be derived as:

$$L_{f(x,y)} = -4f(x,y) + f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) \tag{4.11}$$

The edge detection process was implemented using the following approximation:

$$g(x,y) = f(x,y) * Laplacian \tag{4.12}$$

where the *Laplacian* is defined in Table 4.1.

Sobel and *Prewitt* filters *[Gonzalez and Wintz, 1987]* were also tested on the X-ray images. Their implementation was done by using the kernels presented in Table 4.1.

The difference with the *Laplacian* and *gradient* operators is that the X-ray images were analysed for horizontal and vertical contour in two independent passes and then the results mathematically processed:

$$g_1(x,y) = f(x,y) * Sobel\ horizontal$$
$$g_2(x,y) = f(x,y) * Sobel\ vertical$$
$$g(x,y) = \sqrt{g_1^2(x,y) + g_2^2(x,y)} \qquad (4.13)$$

and

$$g_1(x,y) = f(x,y) * Prewitt\ horizontal$$
$$g_2(x,y) = f(x,y) * Prewitt\ vertical$$
$$g(x,y) = \sqrt{g_1^2(x,y) + g_2^2(x,y)} \qquad (4.14)$$

Results of applying classical edge detection techniques are depicted in Figure 4.13.

Table 4.1 Mask and convolution kernels for classical edge detection and enhancement

x-1,y-1	*x-1,y*	*x-1,y+1*
x,y-1	*x,y*	*x,y+1*
x+1,y-1	*x+1,y*	*x+1,y+1*

3 x 3 Mask

0	-1	0
-1	4	-1
0	-1	0

Laplacian

-1	-1	-1
1	-2	1
1	1	1

Gradient

1	2	1
0	0	0
-1	-2	-1

Sobel Horizontal

1	0	-1
2	0	-2
1	0	-1

Sobel Vertical

-1	-1	-1
0	0	0
1	1	1

Prewitt Horizontal

1	0	-1
1	0	-1
1	0	-1

Prewitt Vertical

Classical methods of image segmentation applied to meat X-ray images with respect to an inspection system have proven to render results that are not very useful for further image analysis such as high-level detection (due to the merging between foreign-bodies and the surrounding background). Since image segmentation output is the input to consequent image processing techniques, one wants that output to be of a high quality. The segmentation process can also be seen as a constraint optimisation problem. The constraints, in this case are based on the fact that objects extracted from

the image needs to be homogenous and different from each other for instance. Thus, an alternative approach to image segmentation was implemented by using a Hopfield Neural Network architecture.

4.4.2 Hopfield Neural Network approach to image segmentation

Image segmentation is viewed as a constraint optimisation problem. A Hopfield Neural Network architecture is proposed for the segmentation of dual-band X-ray images.

4.4.2.1 Hopfield ANN model (HNN)

HNN was proposed in 1985 by Hopfield as a way of solving optimisation problems. In a HNN each neuron is linked to another and weights are symmetrical, i.e. $w_{ij}=w_{ji}$, where w_{ij} represent the weight of connection between neuron i and j. There are no input or output neurons, but rather all the neurons look and act exactly the same. Inputs are applied to all neurons at the same time. There are two versions of HNN: discrete and continuous models. The output function of a neuron is typically a function of the all other weighted neurons inputs.

The network for the optimisation application tends to relax into stable states that minimises an energy function of a Lyapunov form *[Hopfield, 1982], [Hopfield, 1984], [Hopfield, 1985]:*

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} v_i v_j - \sum_{i=1}^{N} I_i v_i \tag{4.15}$$

where N is the number of neurons, v_i is the output of the i^{th} neuron , and I_i is the external input for the i^{th} neuron term. Hopfield demonstrated that HNN relax into a stable state tending to minimise its corresponding energy function. This corresponds to a state of the HNN when no neurons are changing their states in time. For a network with no external input or bias, a simplified energy function can be expressed as:

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} v_i v_j \tag{4.16}$$

Like in the majority of NN, the feedback input to the i^{th} neuron is equal to the weighted sum of neuron outputs v_j ($j=1..N$). The total input to the i^{th} neuron can be expressed as:

$$net_i = \sum w_{ij} v_j + I_i \tag{4.17}$$

The behaviour of the network in time can be determined by differentiating E with respect to v_i, the result in this case being the negative of the updating algorithm:

$$\frac{\partial E}{\partial V_i} = -\sum_{i=1}^{N} w_{ij} v_j \tag{4.18}$$

The minimization of the energy occurs when solving a set of dynamic equations:

$$\frac{\partial v_i}{\partial t} = -\frac{\partial E}{\partial v_i} \tag{4.19}$$

Therefore, the updating algorithm for a neuron i, at a given moment in time t is:

$$v_i^{[t]} = \sum w_{ij} v_j^{[t-1]} \tag{4.20}$$

A general discrete Hopfield Network Algorithm for Optimisation is:

Discrete HNN Optimisation Algorithm

Step 1

Define an energy function based on problem constraints

Compare the energy with the Lyapunov energy function of the Hopfield network (4.15) in order to derive the weights

Step 2

Dynamics of the network

a) Start with random activation levels for the neurons at moment t=0

b) At any time t (t>0)

Compute the output (Activation level for time (t+1)) of each neuron

$$v_i^{(t+1)} = f\left(\sum_i w_{ji} v_i^{(t)} + I_j \right) \tag{4.21}$$

where f is a discrete function defined as follows:

$$f(x) = \begin{cases} 1, & \text{for } x > \eta \\ -1 \text{ or } 0, & \text{for } x < \eta \\ v_j^{(t)} & \text{for } x = \eta \end{cases} \tag{4.22}$$

where η is a predefined threshold.

c) repeat step b) until the activation levels of nodes remain unchanged over the time.

Step 3

Map the final pattern of level of activations with the problem solution pattern.

The continuous model of HNN is based on the real-life neurons behaviour. The neurons have a continuous graded output response and not only a two-state one. The neuron's output function used is:

$$v_i = g_i(\lambda u_i) = \frac{1}{2}(1 + \tanh(\lambda u_i)) \tag{4.23}$$

where u_i is the input to the neuron i and v_i its output. The parameter λ is called the gain parameter and is a constant.

A general implementation of HNN to solve constraint optimisation problems is presented below.

General HNN based algorithm for optimisation problems

1. *determine the input representation in a way that the final solution is indicated by the output of the neurons*
2. *determine the cost function or the energy of the HNN. This should be in the form of a Lyapunov function.*
3. *include constraints terms in the cost function of the HNN (in a Lyapunov form)*
4. *differentiate the defined Energy function with respect of time in order to obtain the equations for the network' dynamics*
5. *randomly choose the starting values for the HNN and start the updating algorithm*
6. *extract the solution from the equilibrum (stable) state of the HNN*

4.4.2.2 The proposed HNN architecture

The segmentation process can also be seen as a constraint optimisation problem. The constraints, in this case are based on the fact that objects extracted from the image

needs to be homogenous and different from each other for instance. Spatial constraints can also be introduced, i.e. objects over the edges of the image are not important. Starting with a random selection of objects, a HNN should be able to reach a stable state, in which all the segmentation constraints are satisfied.

Thus, the strategy used by the majority of the authors comprises two steps: first finding some means of finding a binary representation for the segmentation solution, so that they can be mapped into a HNN stable state; and secondly, choosing the definition of the energy or cost associated function whose minimisation will lead to an optimum solution to the problem.

4.4.2.2.1 Binary representation

It is assumed that the problem is to segment an X-ray image of n by n pixels into k classes, where n is the dimension of the image. The problem is to choose a suitable architecture for the HNN. In this study, the ideas proposed in references *[Armatur et al., 1992], [Koss et al., 1999], [Cheng et al., 1996]* are followed. The solution of the segmentation process using a binary representation can be mapped using a grid of P rows of k neurons. The columns of this architecture represent the classes in which the image has to be segmented. The rows are corresponding to the objects that have to be assigned to a class according to some constraints. Therefore, the number of neurons in such a HNN architecture is:

$$N = P \times k \tag{4.24}$$

If neuron(i,j) is firing (is active), then the objects associated with row i belong to class j $(i=1..P, j=1..k)$.

An approach taken by *[Koss et al., 1999]* is to use a grid of P by k neurons, where P is the total number of pixels in the image:

$$P = n \times n \tag{4.25}$$

Thus, the number of neurons in this approach is $n \times n \times k$. This means that, in the present case, an X-ray image with 536x536 pixels that needs to be segmented into 10 classes would need a HNN with 2,872,960 neurons. The computations associated with the behaviour of such a neural network are very complex and unsuitable for a real

time application. The complexity of such an approach can be decreased severely as in *[Cheng et al.,1996]*. Their HNN consists of a similar grid of N by k neurons, but in this case N is the number of grey-level values found in the input image. The number of neurons decreases dramatically to $N \times k$. In our case, for 10 classes and for all 255 grey-levels values present in the image we only deal with 2550 neurons. This makes this architecture not only manageable from the point of view of computations involved, but also independent of the size of the image.

The architecture of the chosen HNN is depicted in Figure 4.14.

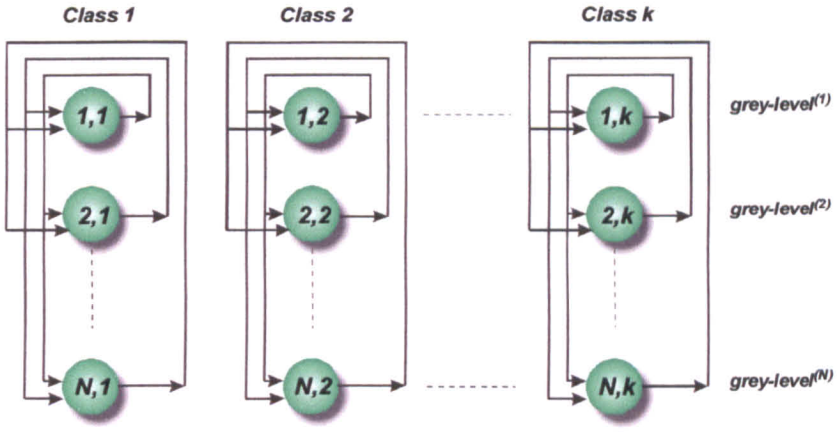
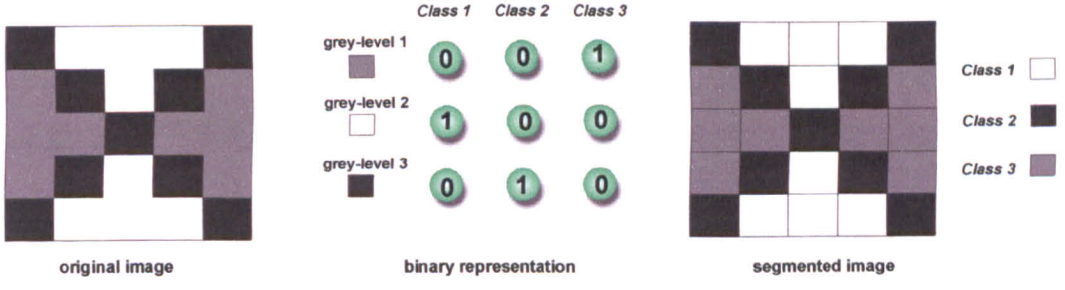
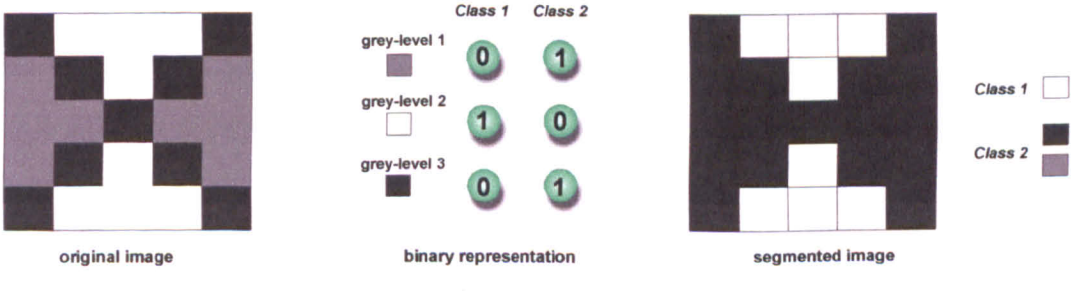


Fig.4.14 HNN architecture



a) Binary representation and segmentation solution for three classes



b) Binary representation and segmentation solution for two classes

Fig.4.15 Binary representation of the solution of segmentation problem

Figure 4.15 described an example of binary representations of an image consisting of 5 by 5 pixels.

As it can be seen from Figure 4.15a, the image that consists of three grey-levels is segmented into three classes. A Hopfield neural-network assigns each grey-level to a class according to a "goodness of segmentation" criteria. If a neuron *(i,j)* is active, then its corresponding grey-level *i* is assigned to class *j*. One grey-level can only be assigned to one class. In Figure 4.15 b, the image has to be segmented only into two classes. Therefore, two grey-levels had to be assigned to the same class.

4.4.2.2.2 Energy definition

Such an energy function must comprise terms for image segmentation constraints $E_{syntactic}$ or syntax energy i.e. to ensure that no grey-level or pixel can belong to two classes at the same times, and terms for goodness of segmentation, $E_{semantic}$ or the semantic energy:

$$E = E_{syntactic} + E_{semantic} \tag{4.26}$$

Using the binary mapping presented above, the segmentation constraints can be summarised as follows:

- Only one neuron per row can be active (output 1); this puts each grey-level into one class:

$$E_{syntactic1} = A\sum_{x=1}^{N}\left(\sum_{i=1}^{k} v_{xi} - 1\right)^2 \tag{4.27}$$

  where A is a constant value

- The sum of outputs of all neurons in one row is 1, this ensuring the fact that each grey-level belongs to one and only one class:

$$E_{syntactic2} = B\sum_{x=1}^{N}\sum_{i=1}^{k}\sum_{\substack{j=1 \\ j\neq i}}^{k} v_{xi}v_{xj} \tag{4.28}$$

  where B is a constant value.

The "goodness" of segmentation has to be measured by the following properties. Firstly, segments have to be uniform and homogenous with respect to grey-level values. Secondly, adjacent regions or segments have to have significant differences with respect to their uniformity (in this case the grey-level values). Thus, the semantic energy is defined in this case as the sum of square distances from each grey-level to the centre of its class. By minimizing the energy, these distances decrease to a minimum leading to a solution for the segmentation. Due to the fact that two images are taken for each product, one high-energy X-ray and one low-energy X-ray image, a semantic energy for both images has to be defined as follows:

$$E_{semantic_1} = C\sum_{x=1}^{N_1}\sum_{y=1}^{N_1}\sum_{i=1}^{k}\frac{1}{\sum_{y=1}^{N} hl_y v_{yi}} v_{xi} DIS_{xy} hl_y v_{yi} \tag{4.29}$$

where N_1 is the number of grey-levels present in the low-energy image, C is a constant and hl_y are the histogram values of the y grey-level for the low-energy band; and

$$E_{semantic_2} = D\sum_{x=1}^{N_2}\sum_{y=1}^{N_2}\sum_{i=1}^{k}\frac{1}{\sum_{y=1}^{N} hh_y v_{yi}} v_{xi} DIS_{xy} hh_y v_{yi} \tag{4.30}$$

where N_2 is the number of grey-levels present in the high-energy image, D is a constant and hh_y are the histogram values of the y grey-level for the high-energy band.

An important aspect in the process of defining the semantic energy is choosing the appropriate measure of distance DIS_{xy}. This represents the distance between grey-level l_x and grey-level l_y. Because the present method is actually a cluster analysis algorithm, a good segmentation can be defined by having spheric or ellipsoidal clusters. The square Mahalanobis distance, originally defined by *[Mahalanobis, 1936]*, is used as a distance measurement. The equation for its calculations is:

$$d_{\Sigma}^2(l_x, l_y) = (l_x - \bar{l}_y)^T \Sigma_y^{-1} (l_x - \bar{l}_y) \tag{4.31}$$

where $\sum_{y}^{-1}$ is the inverted covariance matrix for the class that y grey-level belongs to, l_x and l_y are the data points (the grey-levels) the distance is calculated for, and $\overline{l_y}$ is the centroid vector of the class for grey-level y. If the covariance matrix is the identity matrix, the distance equals the squared Euclidian distance that will allow hyperspherical distribution of clusters:

$$DIS_{x,y} = d^2{}_{l_x,l_y} = (l_x - l_y)^2 \tag{4.32}$$

Thus, the semantic energy can be expressed as:

$$E_{semantic} = E_{semantic_1} + E_{semantic_2} = F \sum_{x=1}^{\max(N_1,N_2)} \sum_{y=1}^{\max(N_1,N_2)} \sum_{i=1}^{k} \frac{1}{\sum_{y=1}^{\max(N_1,N_2)} (hl_y + hh_y)v_{yi}} v_{xi} DIS_{xy} (hl_y + hh_y)v_{yi} \tag{4.33}$$

where F is a constant. As it can be seen from equation (4.33), the histogram values from both low- and high-energy images carry the same weight in the definition of the semantic energy. Experiments were performed with different weightings of the images (ratios of 3/7 and 6/4 were used) with no apparent changes in the results. Thus, in order to decrease the computational overhead, equal weighting of the images was used in the definition of the energy.

Using (4.27), (4.28) and (4.33) into (4.26) one can derive:

$$E = E_{syntactic} + E_{semantic} = A \sum_{x=1}^{N} \left(\sum_{i=1}^{k} v_{xi} - 1 \right)^2 + B \sum_{x=1}^{N} \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} v_{xi} v_{xj} +$$

$$+ F \sum_{x=1}^{N} \sum_{y=1}^{N} \sum_{i=1}^{k} \frac{1}{\sum_{y=1}^{N} (hl_y + hh_y)v_{yi}} v_{xi} DIS_{xy} (hl_y + hh_y)v_{yi} \tag{4.34}$$

A simplification of the energy equation can be done as in *[Armatur et al., 1992], [Cheng et al., 1996], [Koss et al., 1999]* by using a Winner Take All (WTA) scheme. The input-output function for a neuron is modelled as to satisfy the constraints of the energy function. For every row, only one neuron can be active. The neuron that receives maximum input from all other neurons is declared winner and its output is set to 1; the output of the rest of neurons for the same row is set to zero:

$$V_{\substack{x,i \\ x=1..N \\ i=1..k}} = \begin{cases} 1, & \text{if } u_{x,i} = \max\left(v_{\substack{x,i \\ i=1..k}} \right) \\ 0, & \text{otherwise} \end{cases} \tag{4.35}$$

In other words, only one neuron is assigned 100% to a class. This satisfies the syntactic energy terms; therefore the energy equation can be simplified to:

$$E = \sum_{x=1}^{N} \sum_{y=1}^{N} \sum_{i=1}^{k} \frac{1}{\sum_{y=1}^{N} (hl_y + hh_y) V_{yi}} V_{xi} DIS_{xy} (hl_y + hh_y) V_{yi} \tag{4.36}$$

Comparing equation (4.36) with the definition of the Lyapunov energy (4.16), the updating equation for the interconnection weights, when no bias or threshold is present, can be computed:

$$w_{\substack{(x,i)(y,j) \\ x,y=1..N \\ i,j=1..k}} = -\frac{1}{\sum_{y=1}^{N} (hl_y + hh_y) V_{yi}} V_{xi} DIS_{xy} (hl_y + hh_y) V_{yi} \tag{4.37}$$

where V_{xi} and V_{yi} are the binary values for the output of neurons *(x,i)* and *(y,i)*. Because the number of weights that need to be updated is considerable, using the updating formula (4.20) and the weights updating formula (4.37), an equation for the total input to the neuron *(x,i)* can be derived as follows:

$$v_{xi} = -\frac{1}{\sum_{y=1}^{N} (hl_y + hh_y) V_{yi}} \sum_{y=1}^{N} DIS_{xy} (hl_y + hh_y) V_{yi} \tag{4.38}$$

4.4.2.2.3 The HNN based segmentation algorithm

Summarizing the previous paragraphs, the proposed algorithm for the segmentation of dual band X-ray images comprises the following steps:

HNN segmentation algorithm

- *Compute the histogram values for both images hl_i and hh_i;*
- *Compute the initial number of neurons as maximum between the number of grey-levels present in low-energy image and the number of grey-levels present in high-energy image $N=max(N_1,N_2)$;*

- *Compute the distance measure DIST using (4.32)*
- *Initialise the network input values making sure that there is no class with at least one grey-level (one neuron) assigned to it; apply the WTA algorithm (4.35) to compute the new neuron inputs*
- *Repeat*
  - *For each row*
    - *calculate the input of each neuron using (4.38)*
    - *using WTA competitive learning calculate the new input to the neurons (4.35)*
  - *Check convergence – if there are no changing neurons at this moment of time then STOP*
- *Until convergence*
- *Using the final binary values of the neurons assign grey-levels to the corresponding class and write the final output*

The proposed HNN segmentation algorithm is in essence a process of minimizing a cost function defined as the global energy of the network by using the gradient descent method. As in all of the cases, gradient descent search process for the minimum of a function does not guarantee that global minimum will be found (the algorithm can stop when a local minimum is reached and that may not be the best solution). Methods can be applied to solve this problem *[Zurada, 1992]*. However, these are stochastic approaches, that cannot guarantee either that the global minimum will be reached. They are usually based on random searches across the error space i.e. simulated annealing. An attempt to escape local minima was reported in *[Young et al., 1997]* by using a MLHNN. The algorithm suffer from greatly increased computational overhead that would not be beneficial for a real-time application. In the present case, investigations with different random starting configuration for the weights proved to render no significant differences in terms of the segmentation results or of the minimum value of the energy associated with the HNN.

4.4.2.2.4 Experimental investigations

As it was explained in the previous paragraphs, a dual-band image was taken from chicken-breast meat: one high-energy image and one low-energy image from the same product. The segmentation process is difficult due to the fuzziness of the X-ray images. Another factor that makes the segmentation process difficult is the random thickness of the meat. Often, pieces of meat overlap creating a zone of high grey-level masking the bones that might be present in there. Therefore there are a number of investigations required to establish the properties of the proposed method. These are described as follows and mostly use the sample images presented in Figure 4.16.

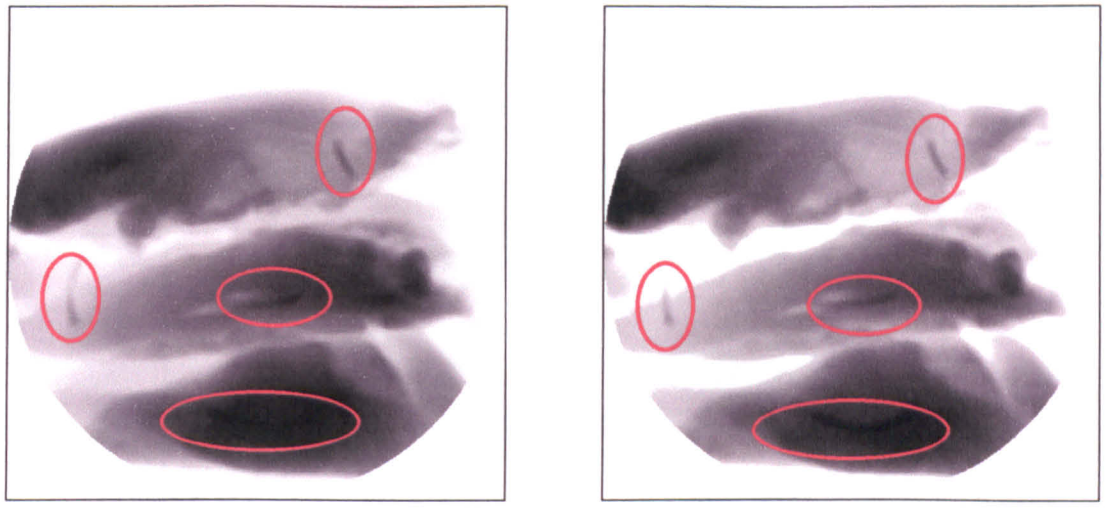


Fig.4.16 Example of the low-energy (left) and high-energy (right) X-ray images for a chicken breast meat product containing 4 bones

4.4.2.2.4.1 Single image segmentation versus dual-band image segmentation

In order to show the advantage of using a dual-band image in the segmentation process (thus, information from two different view of the meat product is used), a HNN was implemented for the segmentation of a single image. In the latter case, the HNN assumes the same architecture, but the associated energy comprises terms for histogram information only from a single image, as:

$$E = \sum_{x=1}^{N} \sum_{y=1}^{N} \sum_{i=1}^{k} \frac{1}{\sum_{y=1}^{N} h_y V_{yi}} V_{xi} DIS_{xy} h_y V_{yi} \tag{4.39}$$

where h_y is the histogram values of grey-level y in the single image.

As it can be seen from Figure 4.17, the results are quite different. When using histogram information from only a single image, some of the bones are merged with the background, therefore making them impossible to be detected by further analysis techniques. In contrast, when using a dual-band based segmentation algorithm, the process makes use of histogram information from both low and high-energy images and the bones are segmented correctly. The algorithm was tested on around 250 single and dual-band images. An improvement was noticed for the latter in around 57.5% of the total cases. Therefore, the advantage of using a dual-band image as opposed to a single image is obvious.

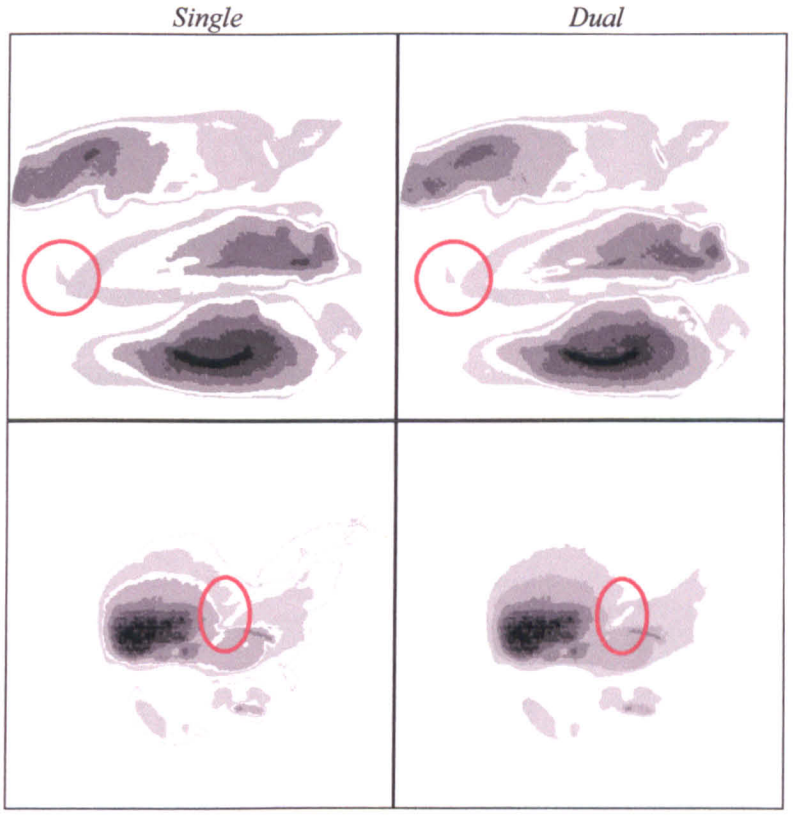


Fig.4.17 Differences between single and dual-band HNN segmentation

4.4.2.2.4.2 Parameter sensitivity

It was said before that the algorithm requires a single parameter: k the number of classes the image needs to be segmented into. Since this number is unknown, experiments were taken for different values of k. The images were segmented into 4,5,6,7,8 and 9 classes. As it can be seen from Figure 4.18, segmentation into 4 and 5 classes does not render the necessary number of objects extracted. Segmentation into 6,7,8 and 9 classes is able to produce good results in terms of the number of objects segmented, but with noticeable differences between the pixels belonging to the same object.

Table 4.2 Average segmentation results for 30 images

Number of classes	Number of correct segmented bones	Percentage of pixels
5	2.8	68.9%
6	3.5	88.4%
7	3.8	89.2%
8	6.8	86.5%

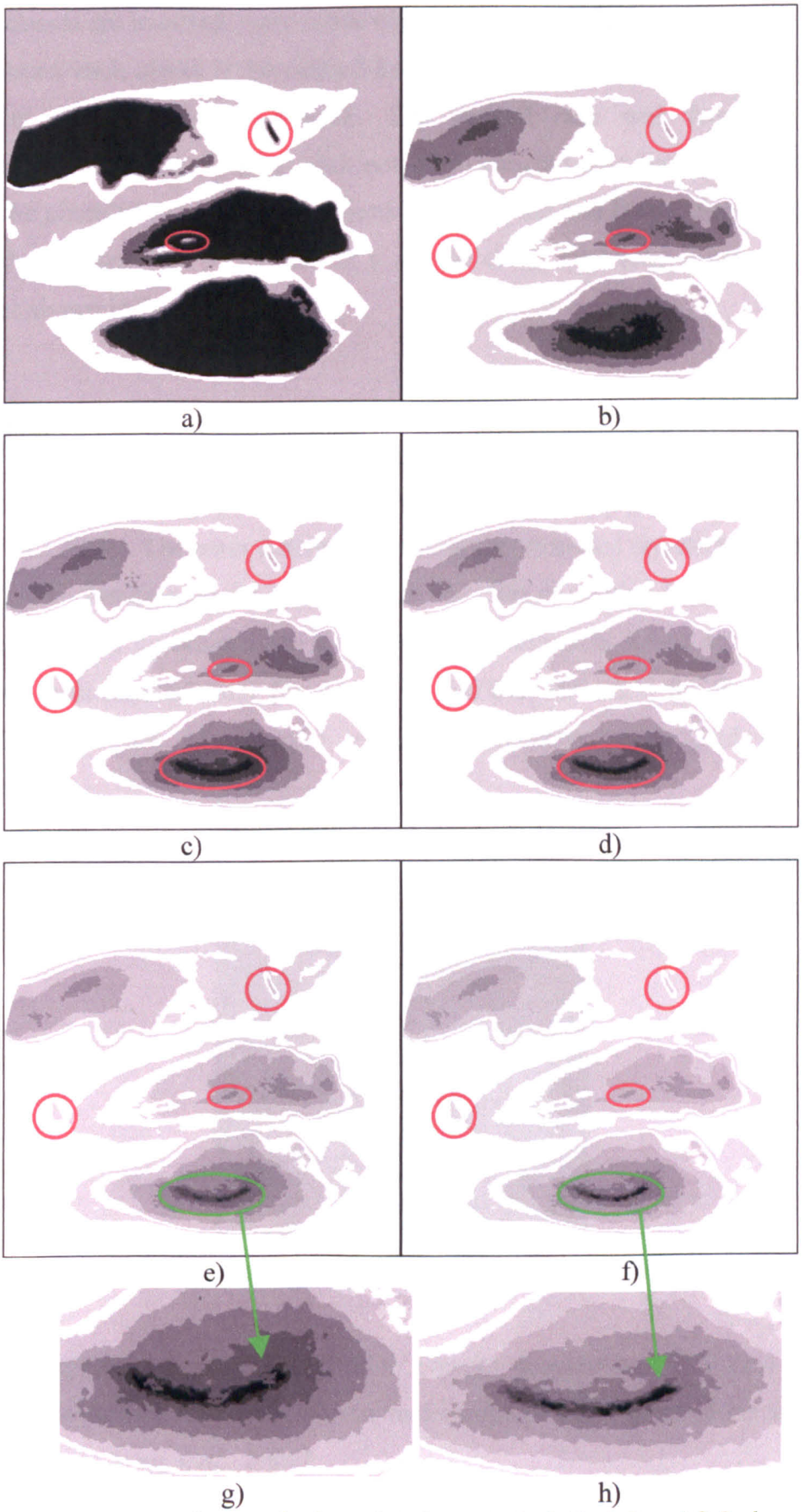


Fig.4.18 Segmentation results into a) 4; b) 5; c) 6; d) 7; e) 8 and f) 9 classes; g), h) fragmentation details for e) and f)

When 6 classes are involved, parts of the objects are merged with the background. For 8 and 9 classes, each object is fragmented into more than 2 pieces, and therefore, makes subsequent processing more difficult. The optimal case was found to be the segmentation into 7 classes. This result not only extracted all 4 bones, but also around 90% of the pixels of the segmented objects belong to the real bones. Segmentation into 5,6,7 and 8 classes was performed on a sample of 30 dual-band X-ray images. The results are shown in Table 4.2.

4.4.2.2.4.3 Comparison with other clustering (multilevel thresholding) techniques

For comparison reasons, another HNN architecture was implemented and tested *[Koss et al., 1999]*. The main difference between the proposed architecture and Koss's approach is in classification the pixels *(x,y)* in one class or another instead of classifying grey-levels (spatial constraints are used instead of histogram based constraints). The results are quite unsatisfactory (Figure 4.19), and the time taken for the algorithm to converge (more than 30 seconds) makes it impossible to be used in real-time applications.

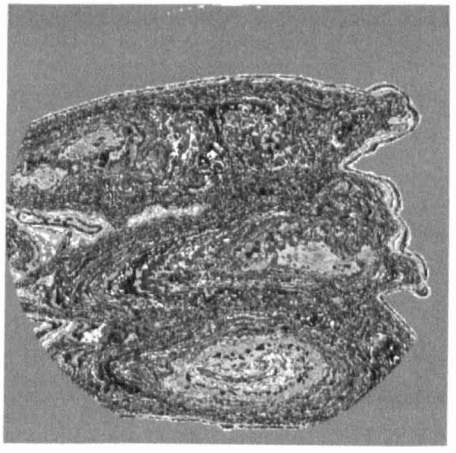


Fig.4.19 Koss HNN segmentation results (7 classes)

Furthermore, the proposed algorithm's results were compared with the fuzzy c-means segmentation algorithm (FCM). An implementation of FCM was done using the following steps *[Cheng et al., 1996]*:

FCM clustering algorithm

1. *the number of classes k for the segmentation needs to be known a priori; randomly choose the initial class centers z_i ; compute the histogram values from the original X-ray image (h_x for grey level g_x); compute the Euclidian distance measure between the grey levels g_x and the class center z_i for all classes:*

$$d_{x,i}^2 = \|g_x - z_i\|^2 \tag{4.40}$$

for $1 \le x \le N$, $1 \le i \le k$

2. *compute the membership matrix U for all pixel as follows*

$$u_{x,i} = \frac{\frac{1}{d_{x,i}^2}}{\sum_{i=1}^{k} \frac{1}{d_{x,i}^2}} \quad \text{for } g_x \neq z_x \tag{4.41}$$

and

$$u_{x,i} = \begin{cases} 1, & \text{if } x = i \\ 0, & \text{otherwise} \end{cases} \tag{4.42}$$

3. *update the class centers as*

$$z_i = \frac{1}{\sum_{x=1}^{N} u_{x,i}^2 h_x} \sum_{x=1}^{N} u_{x,i}^2 g_x h_x \tag{4.43}$$

4. *check convergence of the algorithm:*

$$\max \left\| U^{(t+1)} - U^{(t)} \right\| < \varepsilon \tag{4.44}$$

where $U^{(t+1)}$ and $U^{(t)}$ are the membership matrix at iteration (t+1) and (t) , respectively and ε is a predefined convergence error; if criteria for convergence is reached then stop, otherwise go to step 2.

As it can clearly be seen from Figure 4.20, the results are quite different than when using a HNN. First of all, the fragmentation that appears is quite high and that is mostly due to the fuzziness of the original X-ray images. Secondly, the FCM

segmentation algorithm does not extract all the bones, maximum performance being 2 out of 4 bones.

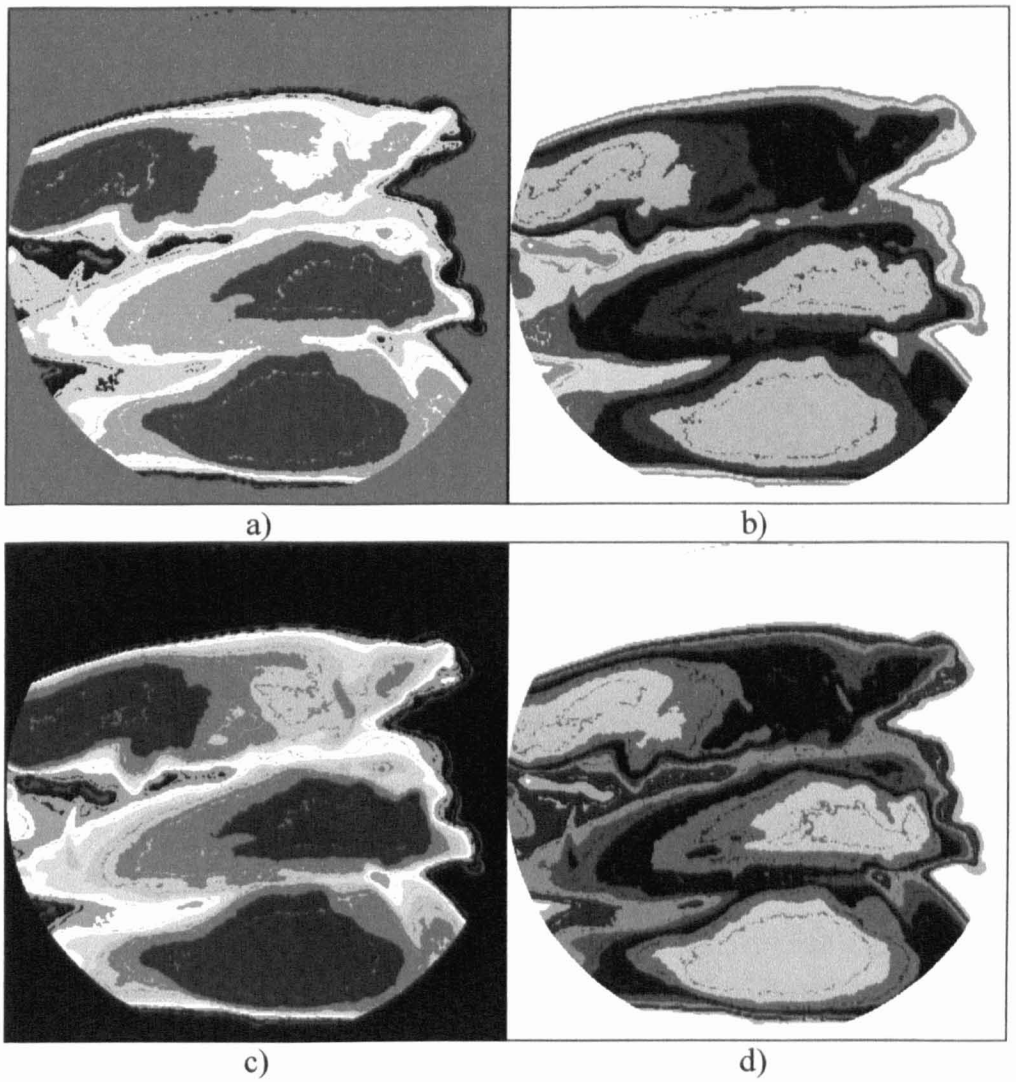


Fig.4.20 FCM segmentation results for a) 5 classes; b) 6 classes; c) 7 classes; d) 8 classes

4.4.2.2.4.4 Noise concerns

As explained previously, due to the nature and design of the X-ray inspection system used, noise can accidentally appear in the input images mainly because of the analogue video signal being transmitting over a coaxial cable from the video camera to the computer interface. In this case, the noise is called "salt and pepper" or impulse noise and can affect pixels in a random manner. When a pixel is affected, its grey-level will change randomly with a random percentage (i.e. 100% for black and white pixels).

Furthermore, geometric distortions ("pincushion" or "barrel" distortion) can appear due to the image intensifier present in the system (see Appendix A). One can

consider these unwanted geometric transformations of the input images as being noise. However, these geometrical effects will not influence the output of the proposed segmentation algorithm since they do not involve some sort of major histogram changes.

Table 4.3 The simulated added noise

Case	Level of noise	Percentage of pixels affected
1	100	5
2	100	10
3	100	30
4	50	5
5	50	10
6	50	30
7	25	5
8	25	10
9	25	30
10	10	10

The image intensifier and the radial nature of the X-ray source can also induce the "vignetting" effect (images look brighter in the middle and darker at the edges). This obviously can be treated as noise since it affects the histogram values for the input images.

Since the impulse or "salt and pepper" noise is removed in the pre-processing stage of the system, the only cause for its reappearance is a hardware error. Noise was simulated using two parameters: the level noise which is the percentage with which each pixel's grey-value is affected and the percentage of pixels affected by it from the total number of pixels in the X-ray image:

$$f_n(x,y) = f(x,y) + n(x,y) \tag{4.45}$$

where $f_n(x,y)$ is the noise corrupted image with $n(x,y)$ additive random noise.

It is assumed that there is no change in the noise patterns for the dual-band X-ray image. This means that the low- and high-energy images are corrupted in the same manner (the same pixels are affected with the same amount of noise). Table 4.3 illustrates the parameter values for the simulated noise.

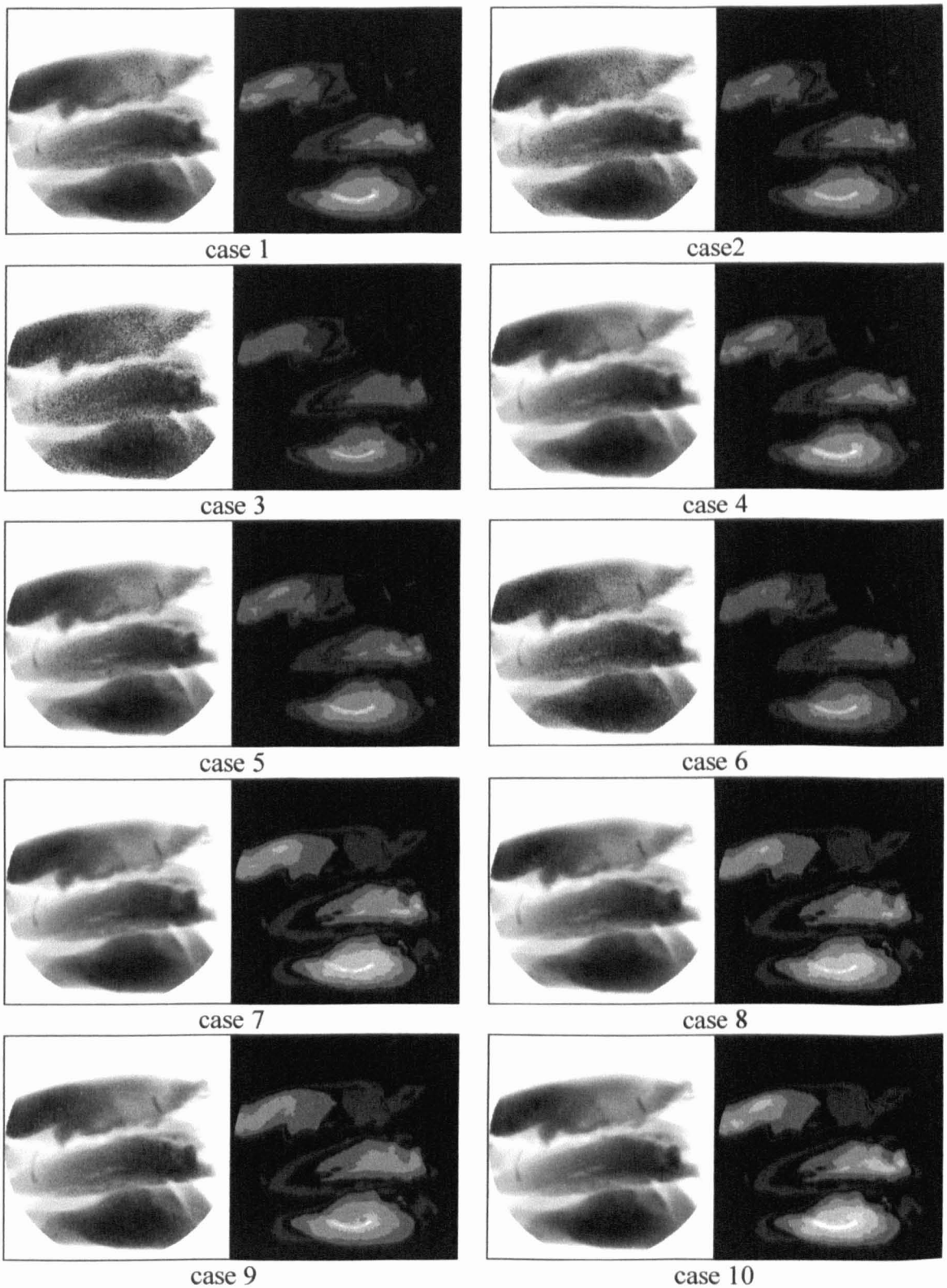


Fig. 4.21 Noise corrupted images and corresponding segmentation results

Cases 1,2 and 3 are considered very extreme cases and their occurrence probability is next to nothing; cases 4,5 and 6 are of a more higher occurrence probability whereas cases 7,8,9 are mostly to appear in real life. The last case (10) is

considered as being the most probable real noise corruption that can appear. The results, as depicted in Figure 4.21, have shown to change dramatically in cases 3 and 6. Even though some foreign bodies (bones) were merged with the background, there are still bones extracted correctly. Furthermore, some fragmentation of the extracted objects appears in cases 2,4,5 and 7, but all bones (or part of bones) are extracted correctly. The segmentation results remained unchanged for the last two cases, (similar to real possible situations).

An X-ray image corrupted with vignetting effect generated noise is presented in Figure 4.22. The segmentation result is proven to be unchanged. Tests were performed on a batch of 30 corrupted X-ray image. No significant changes appeared in the segmentation results.

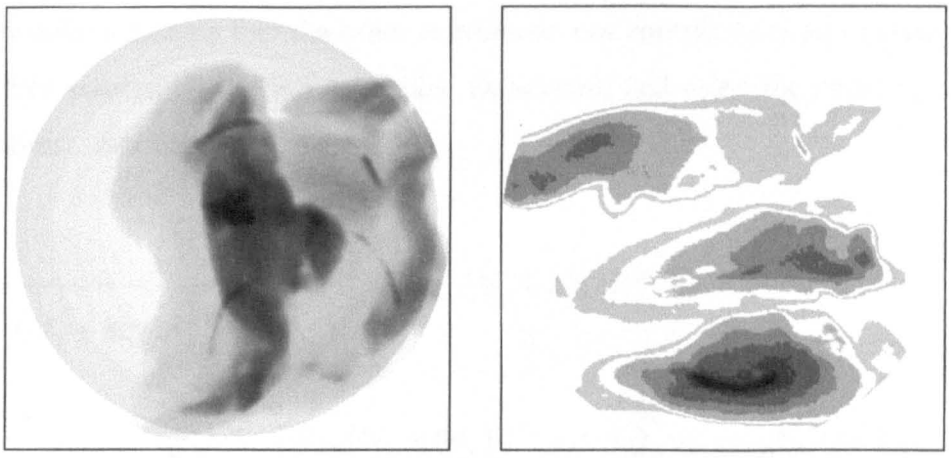


Fig.4.22 Segmentation results in the presence of vignetting effect

Thus, as a general conclusion, it can be seen that the algorithm is performing well even in the presence of specific noise and it is robust.

4.4.2.2.4.5 Convergence of the proposed algorithm

To demonstrate the convergence of the proposed algorithm, it needs to be demonstrated that:

$$\lim_{t\to\infty}(E^{(t+1)} - E^{(t)}) = 0 \tag{4.46}$$

In other words, the difference between the energy value at time *(t+1)* and the energy at time (*t*) is decreasing over the time. Supposing that neuron *q* situated on row *r* is the only one changing its state from time *t* to time *t+1*, the change to the energy level ΔE associated with the HNN is :

$$\Delta E = E^{(t+1)} - E^{(t)} = \sum_{x=1}^{N} \sum_{y=1}^{N} \sum_{i=1}^{k} \frac{1}{\sum_{y=1}^{N} (hl_y + hh_y) V^{(t+1)}{}_{yi}} V^{(t+1)}{}_{xi} DIS_{xy} (hl_y + hh_y) V^{(t+1)}{}_{yi} -$$

$$\sum_{x=1}^{N} \sum_{y=1}^{N} \sum_{i=1}^{k} \frac{1}{\sum_{y=1}^{N} (hl_y + hh_y) V^{(t)}{}_{yi}} V_{xi}{}^{(t)} DIS_{xy} (hl_y + hh_y) V^{(t)}{}_{yi} \tag{4.47}$$

Since *hl, hh* and *DIS* are constant over time, and only neuron *q* is changing its state, it is safe to assume that the other neurons do not contribute at all to changes in the energy (fact assured by the WTA learning algorithm), and using the updating algorithm (5), equation can be rewritten as:

$$\Delta E = \frac{1}{\sum_{y=1}^{N} (hl_y + hh_y) V^{(t+1)}{}_{yq}} \sum_{i=1}^{k} V^{(t+1)}{}_{ri} DIS_{ry} (hl_y + hh_y) V^{(t+1)}{}_{yq} -$$

$$\frac{1}{\sum_{y=1}^{N} (hl_y + hh_y) V^{(t)}{}_{yq}} \sum_{i=1}^{k} V^{(t)}{}_{ri} DIS_{ry} (hl_y + hh_y) V^{(t)}{}_{yq} = -\left(\sum_{y=1}^{N} w_{(r,i),(y,q)} V_{ri} \right) \cdot \acute{C} V_{yq} \tag{4.48}$$

where

$$\Delta V_{yq} = V_{yq}^{(t+1)} - V_{yq}^{(t)} \tag{4.49}$$

and the first sum term is the total input to neuron *k*.

The WTA algorithm also assures that for each row, only one neuron's output is 1 and the rest outputs are set to 0, therefore there can exist only three cases:

- $\Delta V_{yq}=-1$
- $\Delta V_{yq}=0$
- $\Delta V_{yq}=1$ (4.50)

In the first case, neuron q changes its state from 0 to 1 and its total input must be greater than its threshold. In the second case, neuron q changes its state from 1 to 0 and therefore its total input must be less than its threshold. In the last case, $\Delta E=0$. This shows that in all cases $\Delta E<0$ and therefore the energy does not increase overtime whenever a neuron changes its state. Thus, the proposed HNN will eventually reach a minimum energy state and the algorithm will stop.

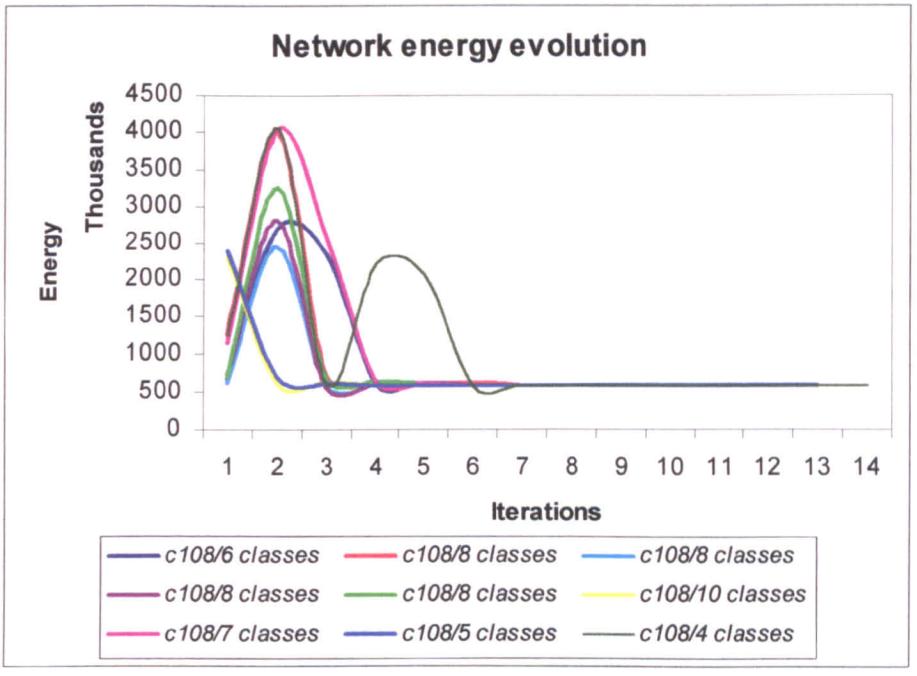


Fig.4.23 Network energy evolution in time

The experimental results are backing up the theoretical findings. Since the network starts with random clusters, the energy fluctuates in the first iterations and then tends to decrease until it is stabilised at a minimum value. The number of changed nodes rapidly decreases with every iteration, thus supporting the theory. The energy evolution for the segmentation of an X-ray dual-band image in a different number of classes is depicted in Figure 4.23. The corresponding number of changed nodes is illustrated in Figure 4.24.

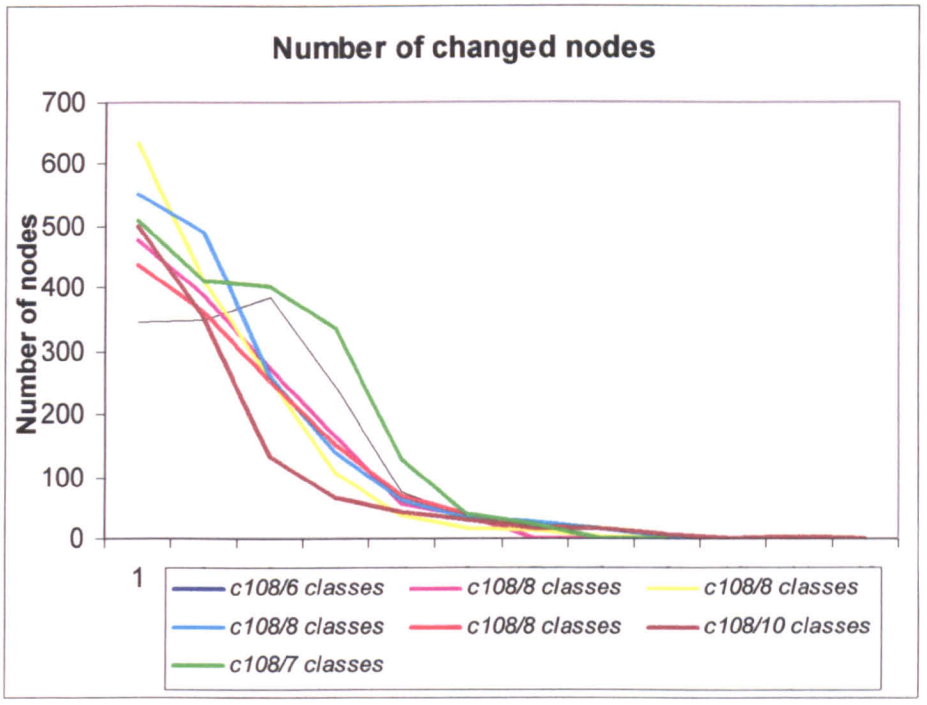


Fig.4.24 Number of changed nodes versus number of iterations

4.4.2.2.4.6 Real-time concerns

Since the inspection system needs to be used in a real-time environment, time is a very important issue. Each part of the system has to perform in the smallest amount of time possible. Since HNN architecture is one of the main components of the inspection system, it too has to be time efficient. A series of tests regarding time consuming were performed. The computations were measured on an Intel Pentium III, 1 GHz processor, but at the current time of the print, system with as much as 1.7 GHz appeared. Also, it has to be remembered that with the growing developing rate of new faster and more powerful processors, the algorithm performance with regards to the time consumption will exponentially increase.

Table 4.4 Time[sec.] for different images

Image number	Segmentation into 5 classes	Segmentation into 6 classes	Segmentation into 7 classes	Segmentation into 8 classes
108(fig.)	0.15	0.19	0.24	0.3
64(fig.)	0.15	0.21	0.38	0.4
329(fig.)	0.11	0.19	0.27	0.46
321(fig.)	0.15	0.19	0.22	0.46
429(App.)	0.13	0.23	0.68	0.82
430(App.)	0.12	0.16	0.2	0.27
109(App.)	0.13	0.18	0.22	0.3

As it is obvious (see Table 4.4 and Figure 4.25), time required for the algorithm increases with the number of classes for the segmentation. No significant differences were observed when the HNN segmentation algorithm was tested on a sample number of images for the same number of classes. The differences ranged from 0.03 seconds to 0.02 seconds, and these values are considered unimportant even for a real-time inspection system (see Figure 4.26).

Some variations exist even when segmenting the same image into the same number of classes and this is mostly due to the randomly chosen starting values of the HNN architecture. However, the differences are small and can be regarded as insignificant to the computed average time elapsed value.

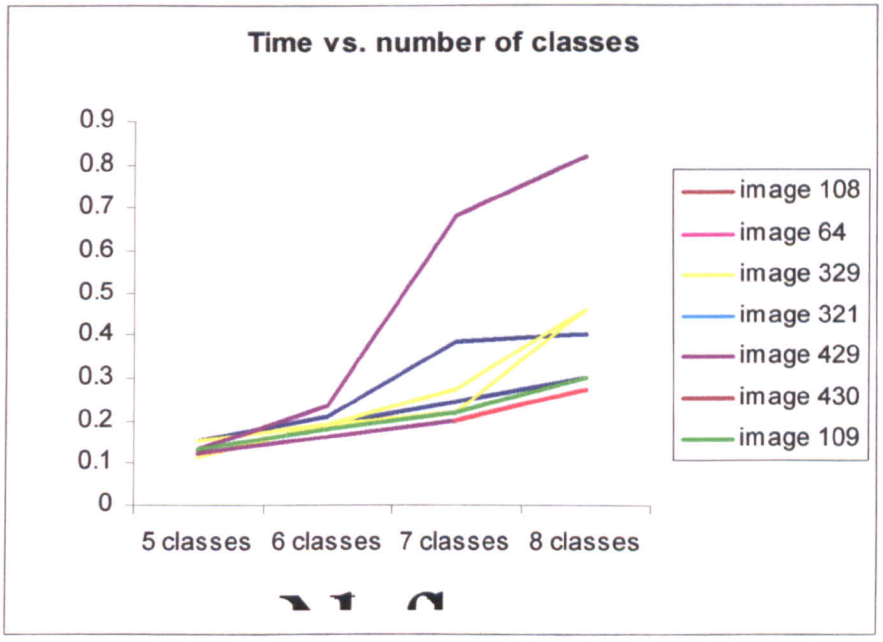


Fig. 4.25 Time increase with the number of classes needed for segmentation

An observation needs to be presented here. Even though, time elapsed computations were performed here, they are of a relative importance. When using a computer visual programming environment, such as Borland Delphi, inherent time increases occur due to the nature of the environment (to present as much visual cues as possible to the end user). The main resource – the processor- is also allocated to other processes running at the same time in the PC such as updating the real-time clock of the system or interrupt requests from other hardware devices, and these lead to other incurring time increases. However, when dealing with a dedicated hardware designed

specifically for HNN X-ray image segmentation (using massive parallel hardware VLSI) the time can be decreased considerably. The scope of the thesis does not include the optimisation of the programming code from the time point of view.

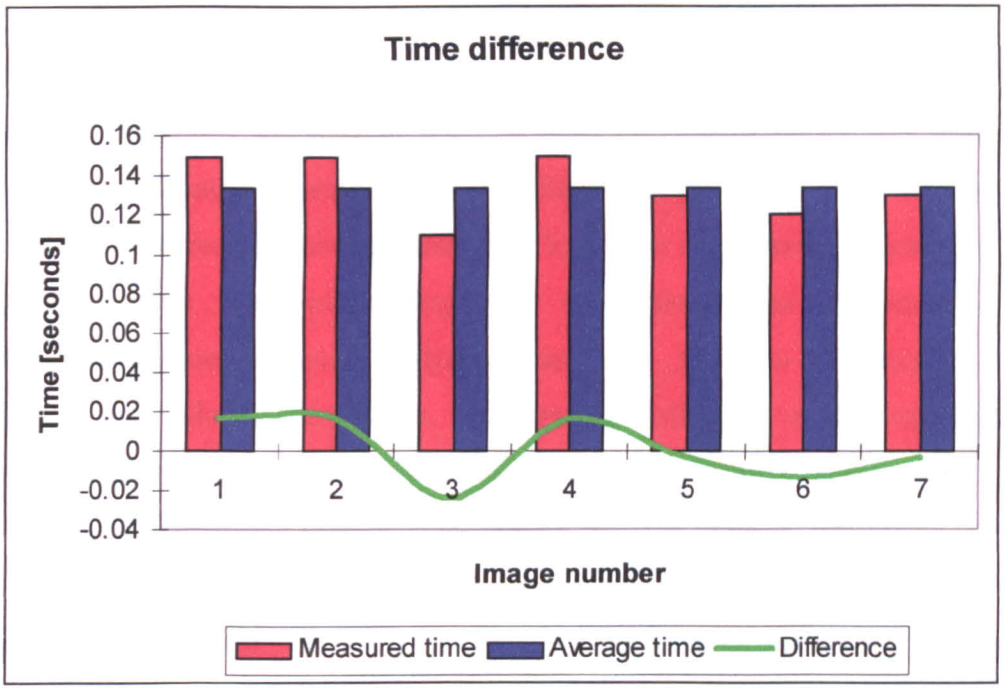


Fig.4.26 Time difference across a range of images for segmentation into same number of classes

The performed experiments have proven that the proposed HNN based segmentation algorithm is robust, fast and it renders good results with regards to the segmentation process. Now, the segmented image needs to be analysed and the partitioned objects classified as foreign-bodies or not.

4.5 Post processing and high-level detection

Once the segmentation process is over, the resulting segments or objects need to be analysed and a decision has to be taken whether they are foreign bodies or not. This is a process of classifying the objects as being foreign bodies or not. "*The process of identifying objects in images is called pattern recognition*" *[Nadler and Smith, 1993].* Pattern recognition (PR) is the research area that studies the operation and design of systems that recognize patterns in data. It contains subdisciplines like discriminant analysis, feature extraction, error estimation, cluster analysis (together sometimes called 'statistical' pattern recognition), grammatical inference and parsing (sometimes called 'syntactical' pattern recognition). Important application areas are image analysis, character recognition, speech analysis, man and machine diagnostics, person identification and industrial inspection. A pattern is defined as an "arrangement or an ordering in which some organisation of underlying structure can be said to exist". *[Watanabe, 1985]* defines a pattern as an "entity, vaguely defined, that could be given a name". Pattern recognition involves "the partitioning or assignment of measurement, experimental data, stimuli, or input patterns into meaningful categories" *[Watanabe, 1985].* It naturally involves extraction of significant attributes of the data from the background of irrelevant details. For example, in character recognition, a matrix of pixels (or strokes) is mapped into characters and words. Other examples include: signature verification, recognition of faces from a pixel map, identification systems, etc. This approach can be applied to the inspection systems in food industry. Usually the input is the image acquired from an inspected product and the output can be the location of bones or foreign bodies within that product. In this case the pattern recognition system will involve a mapping process into bone or non-bone categories.

Practical image recognition systems generally contain several stages in addition to the recognition itself. Before moving on to focus on the specific recognition case presented here, a typical recognition system will be described. A general pattern recognition system can comprise a feature extraction unit, a classifier unit and a context processor (see Figure 4.27 *[Chen, 1973]*). From the input objects or patterns, the feature unit extracts information in order to facilitate recognition. The classifier unit identifies which pattern the object belongs to. The context processor increases recognition accuracy by providing relevant information regarding the environment surrounding the object.

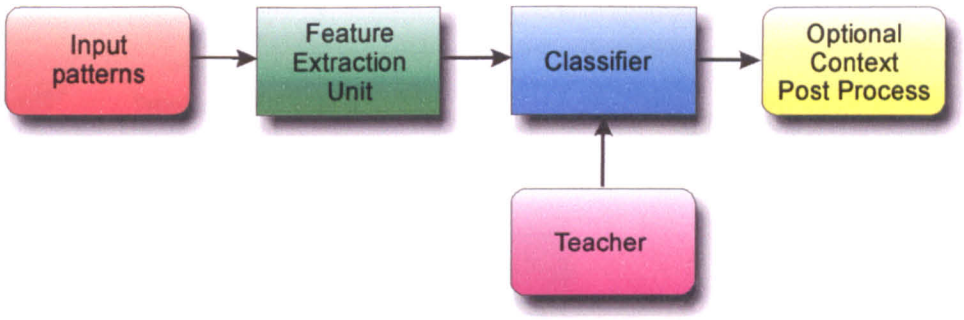


Fig. 4.27 Components of a pattern recognition system *[Chen, 1973]*

Such a system is expected to perform either supervised classification (where a given pattern has to be identified as a member of already known or defined classes) or unsupervised classification (where a pattern needs to be assigned to a so far unknown class of patterns). The task of pattern recognition may be complicated when classes overlap. In this case the recognition system must attempt to minimise error due to misclassification.

There are two major approaches to designing a pattern recognition system: statistical and syntactic (or structural). Some authors regard the PR methodologies as classical or AI-based. In the first case, classical statistical or mathematical methods are employed, whereas in the latter case, AI techniques such as NNs or fuzzy logic are used as the core of the PR classifying process.

In the present case, the analysis of the segmented image can be regarded as a PR problem. Segmented images contain the patterns that need to be classified or recognised. Since the segmentation was performed using seven levels, the resulting partitioned image has objects characterised by seven grey levels. The methodology followed is presented in Figure 4.28.

Firstly, for each segmentation level, an object is extracted. Then, for that object features are extracted. Finally, based on the features extracted from the object, this object is classified either a foreign body (i.e. bone) or normal meat. Another object is then extracted and the procedure is repeated for all objects and for all segmentation classes. However, at any moment, if an object is found to be a foreign body, the algorithm stops, and the product is rejected.

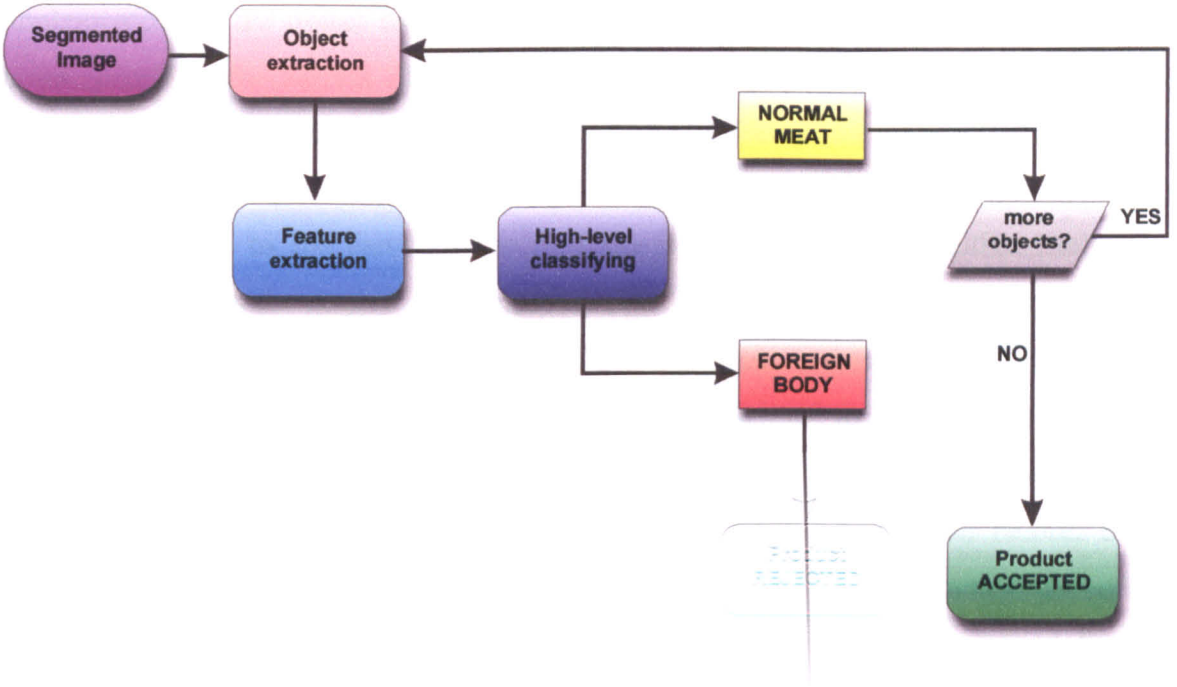


Fig. 4.28 Summary of the foreign-body detection system

In the following paragraphs, classical and AI-based PR will be presented along with their advantages and shortcomings. A critical comparison between methods is presented.

4.5.1 Object extraction and feature extraction

4.5.1.1 Object extraction

The segmented image, as explained in the previous paragraph has seven levels of discrimination. Each level may contain one or more objects. An object is defined as being a contiguous area that contains pixels of same grey level values (unified by the segmentation process as a single value from 1 to 7). Since in most of the cases, each level contains more than one object, there is a need for an object extraction process. The object extraction process extracts the objects from a segmentation level one by one.

For each segmentation level, a backtracking algorithm was designed and implemented for the object extraction. It consists of a function that requires a starting point (pixel) of the current object to extract. Then, all pixels adjacent to the starting pixel are tested for similarity. If a pixel is found to be the same grey-level as the object, then it is marked respectively. Otherwise the process is stopped. The algorithm is called backtracking due to the fact that the function is called recursively within itself. The algorithm searches for neighbour pixels from right to left for instance and it continues to do so until a dead end is reached; then the function "backtracks" its way to the starting point and the process continues for considering pixels from left to right and so on.

When the function has marked all the pixels of an object, features are extracted from it and further high-level classification is performed. As explained before, if the object analysed is found to be a foreign body, then the entire inspection process is stopped and the product is rejected. Otherwise, the object is simply erased for the analysed class and another one is extracted using the same backtracking technique.

Backtracking algorithm for object extraction

```
function Object_extraction(starting point(x,y))
   begin
            if starting_point(x,y)
                        (does not belong to the object segmentation
                     level)
                  and (it was considered before)
                  then STOP
            else
                  //Pixel belongs to the object and it was not
                  //considered before
                  begin
                        mark (x,y) as (already being considered);
                        //start and check the neighbouring pixels
                        Object_extraction(x+1,y);
                        Object_extraction(x-1,y);
                        Object_extraction(x,y-1);
                        Object_extraction(x,y+1);
                  end;
end;
```

The starting point of an object is obtained by scanning the segmented image line by line until the first pixel within the current segmentation level is found. That pixel is considered as being the starting point of a new and not considered object within the analysed class. Examples of object extraction are presented in Figure 4.29. Seven objects are presented within that image segmentation level.

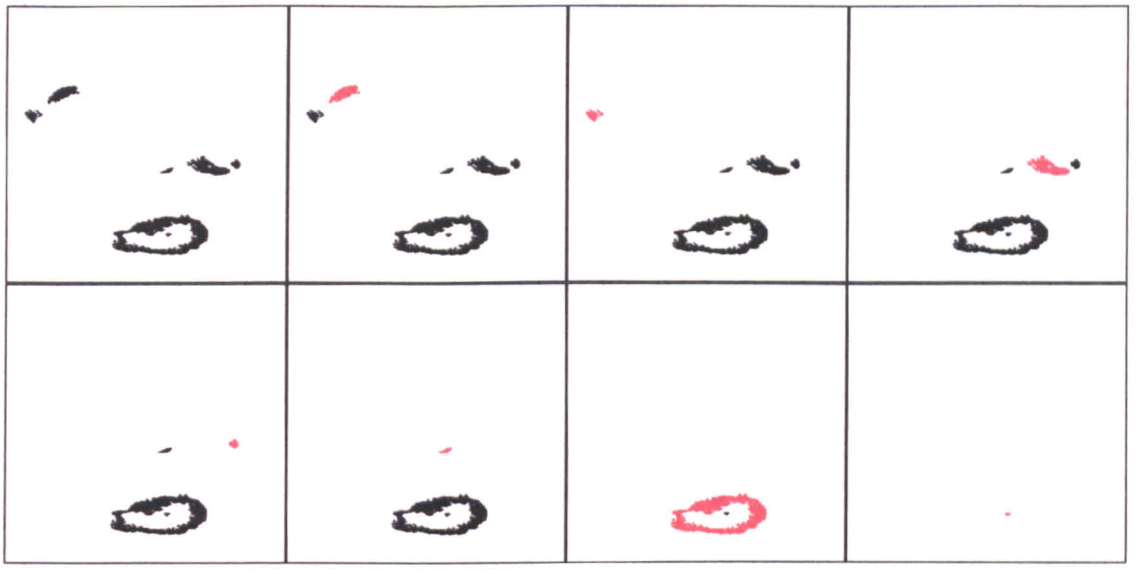


Fig. 4.29 Object extraction methodology for image 4.

4.5.1.2 Feature extraction

The end result of the segmentation and object extraction is an array of numbers that represents the object of interest in some way (in the present case a list of coordinates). In most of the practical cases, there may be just too many of these numbers and many or most of them would not help to distinguish the class that the objects belongs to from other classes. This is where a feature extraction process is necessary.

Feature extraction can serve for achieving two different goals. Firstly, a reduction of dimensionality of the input pattern space is achieved. Secondly, the feature extraction process will render the most suitable characteristics (features) for the consequent decision process.

Classical mathematical methods were employed in the present case for computing the geometrical or logical features. Due to the implementation of three different PR systems (look-up table, NN-based and fuzzy logic based systems), the feature extraction process is a context specific methodology. However, there are common features for all three approaches. Among these features, the following were computed:

- The physical area size of the current object (measured in pixels) AREA_SIZE, variable that can be regarded as the total number of pixels in the current object;

- The perimeter of the current area PERIMETER: which is the number of pixels n the edge that surround the object
- The average grey-level intensity of the object BRIGHTNESS, computed as follows

$$BRIGHTNESS_{object} = \frac{\sum_{i=1}^{AREA\_SIZE} f(x,y)}{AREA\_SIZE} \quad , (x,y) \in object \tag{4.51}$$

where $f(x,y)$ is the grey-level value or intensity of pixel (x,y);

- The difference in intensity between the object and a rectangular area that surrounds it - DIFFERENCE; a surrounding rectangle was defined for computing the difference measure as can be seen on Figure 4.30.

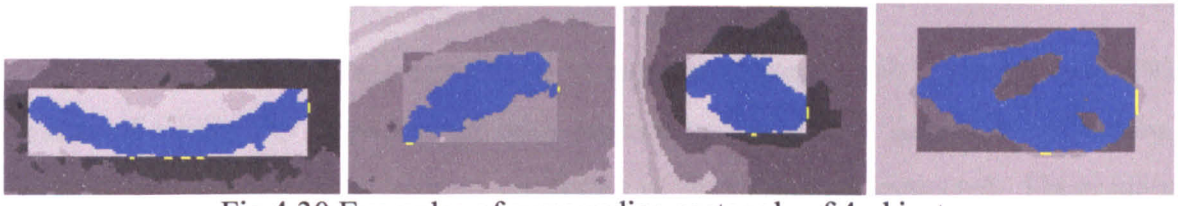


Fig.4.30 Examples of surrounding rectangle of 4 objects

- The physical size (in pixels) of gaps embedded within objects, referred to as HOLE; examples of objects that contain holes are presented in Figure 4.31;

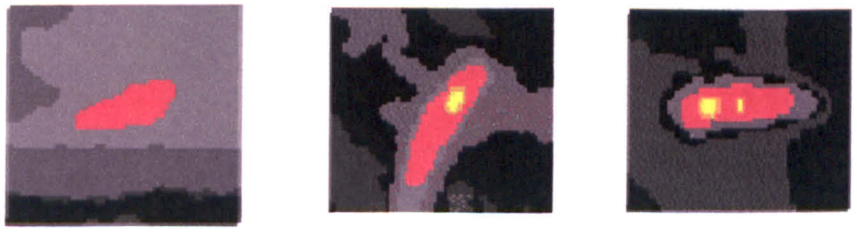


Fig.4.31 Example of objects without holes (left), with one hole (middle) and with two holes (right)

Other features were extracted wherever there was a need for more precise characterisation of the objects. Once the features are extracted, they need to be analysed and compared with stored or known patterns in order to take a final decision. Thus, high-level classification is required, and the process will be presented in the following paragraphs.

4.5.2 Classification

The classification, or decision, process is intended to carry out the final classification of the unknown objects. Once a specific design method has been selected, one is still faced with the actual design and implementation problem. In some cases, representative patterns of each class under consideration are available; thus supervised PR techniques are applicable. This involves the use of a learning environment, where the system is taught to recognize patterns by means of various schemes. Essential for this approach are a number of sets of known patterns for classification. As opposed to the supervised PR techniques, the unsupervised PR methods are used in the situations when only unknown pattern classification is present. The system must learn the pattern classes present in the given data, process also known as "learning without a teacher".

Three different supervised approaches to the classification process will be proposed and presented in the following paragraphs.

The features were extracted from 50 segmented images; this rendered about 785 objects of which 135 were foreign body regions and 650 were normal meat regions. From all 785 objects, input vectors consisting of features were extracted. These input vectors were divided into training and testing data as presented in Table 4.5, vectors needed for training and testing the proposed architectures.

Table 4.5 Training and testing vectors used in the following paragraphs

	Foreign body data	Normal meat data	Number of cases
Training vectors	108	520	628
Testing vectors	27	130	157

The designer should very carefully select about the optimal values for the system he is designing, in order to obtain an acceptable overall performance. This is mainly dependent on the field in which the pattern recognition system will be used. In the case of quality control in the pharmaceutical industry, the goal is to obtain a high rate of true positives (correct prediction of foreign bodies) no matter how many false positives (normal objects classified as being foreign bodies) are produced. In the present case, in which the foreign bodies from chicken meat has to be detected (if any), those values should be specified by the final user. There may be factories that do not want to

miss a single foreign body (such as in the baby food industry) and here the false positive rate is not important. At the same time, there may be factories that are only interested in a few foreign bodies (big bones) and it really does not matter if a few bones are skipped and here the false positives rate has to be a low value.

A well-designed PR system should be flexible and should take into account all those factors. It is the intent of these approaches to offer a flexible alternative to the traditional way of design, in the sense that the architecture will remain the same, no matter where it will be applied, and only a few parameters may be changed in order to achieve robust and flexible functionality. Thus, for the present case, features were divided into four flexibility classes or categories, as presented in Table 4.6.

Table 4.6 Flexibility classes

Data Vectors	ALL	SMALL	MEDIUM	LARGE
Foreign-body	157	35	98	24

Category ALL contains all the foreign-bodies, no matter what their physical properties are. The other three categories were divided according to the foreign-body physical size: SMALL, MEDIUM and LARGE. One can be interested in only rejecting or finding MEDIUM foreign-body, whereas another can be looking for ALL foreign-bodies within a meat product.

4.5.2.1 Classical approach – lookup table

Once the features are extracted from an object, the simplest way is to compare them with the information already stocked within a knowledge database (or a simple look-up table). The knowledge base is usually designed using expert observations. Sample images are analysed by human experts and the information is saved in a useful format in an expert database. The methodology is presented in Figure 4.32. Five features were used for each pattern in this classical approach to PR:

- BRIGHTNESS – average X-ray intensity of the region;
- DIFFERENCE – difference between BRIGHTNESS and the average intensity within a surrounding rectangular area;
- AREA_SIZE – the physical area of the region;
- PERIMETER – the physical perimeter of the region;

- HOLE – the physical size of holes or gaps within the object.

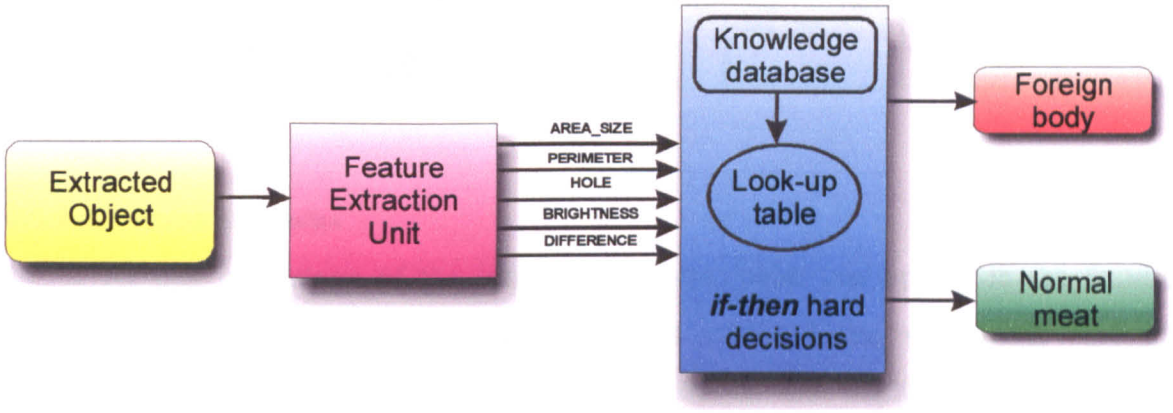


Fig.4.32 Classical approach to PR using a look-up table

The implemented look-up table consists of already stored patterns defined using hard ***if-then*** decisions. The main criteria for a foreign-body, extracted from the expert knowledge database that was gathered through experiments, is defined as summarised in Table 4.7.

Table 4.7 Foreign body definition

Feature	Hard condition
AREA_SIZE	Between 50 and 1300 pixels
HOLE	Between 0 and 50 pixels
PERIMETER	Between 0 and 300 pixels
BRIGHTNESS	Between 180 and 250 (grey level values)
DIFFERENCE	Between 230 and 255 (grey level values)

Objects with any other characteristics or features are considered as being normal meat.

4.5.2.1.1 Experimental investigations

Flexibility classes (4.5.2) were defined in order to meet end-user expectations on a broad range of applications. Four classes were defined: ALL, SMALL, MEDIUM and LARGE. The result of such an approach is presented in Table 4.8.

Table 4.8 Performance of a classical approach

	ALL	SMALL	MEDIUM	LARGE
PERFORMANCE (true positives)	86.3%	83.6%	90.2%	87.2%

4.5.2.1.2 Noise concerns

At this later stage of the inspection system, noise can appear in the feature extraction process, therefore altering a feature's values. Using simulation, different levels of noise were used to corrupt the input vectors. The method used consists of modifying the input vectors with a predefined percentage called noise level. That means that all values of all testing vectors were modified with a fixed percentage. If the noise level was set at 5%, then all test vectors values were modified either with a 5% increment or a 5% decrement. The same process was applied for different values for the level of noise.

Table 4.9 Noise levels and the classifier's corresponding performance

Noise level	3%	5%	10%	15%	20%	25%	30%
Performance (true positives)	85.2%	72.3%	64.6%	59.1%	50.3%	41.2%	32.7%

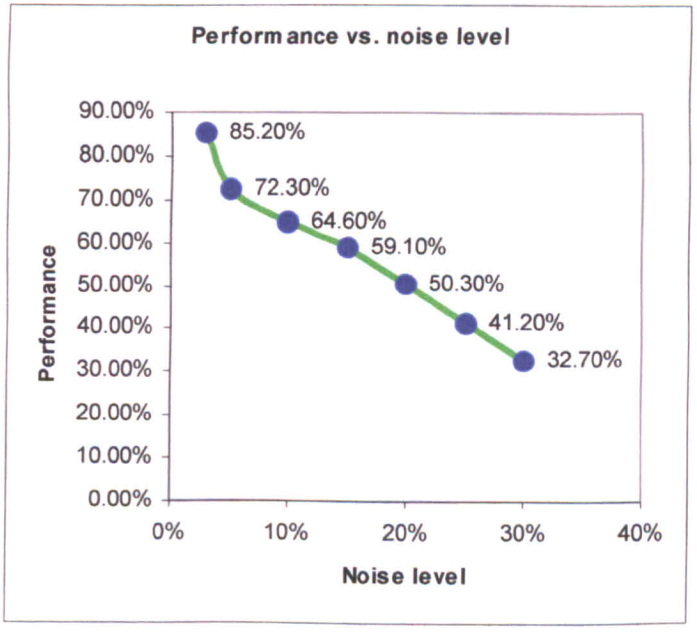


Fig.4.33 Performance affected by noise

As expected, when the noise level is quite low, the performance remains almost unchanged. However, when the noise level increases, the performance decreases exponentially. The results are depicted in Table 4.9 and Figure 4.33. Thus, the architecture is tolerant to noise.

4.5.2.2 Neural network approach

Multilayer perceptron (MLP) networks trained by back propagation *[Rummelhart et.al, 1986]* are among the most popular and versatile form of neural networks classifiers. It has been shown that multilayer perceptron networks with a single hidden layer and a non-linear activation function are universal classifiers *[Funahashi, 1989], [Hartman et al., 1990], [Hornik et al., 1989]*. Thus given an input vector consisting of a set of features representing an input pattern, this network is known to have the discriminative power necessary to serve as a strong recognition engine.

Furthermore, a NN-based PR system will inherit some of the neural networks characteristics such as graceful degradation in the presence of noise, ability to generalise and potential for parallel processing.

In the problem of detection of foreign bodies within food products, a neural network can be used *[Papanicolopoulos et al., 1995]* as a pattern recognition system. The network could be trained with known values for the R/C ratio and then used to recognise the type of foreign body within the inspected product. One of the advantages will be the total replacement of the human operator. Furthermore, such a system will be able to pinpoint the location of detected bones and also to perform really well when one of the intensities of the scattering fields is not read accurately (noisy input).

A simplified scheme of a neural network pattern recognition system is presented in Figure 4.34.

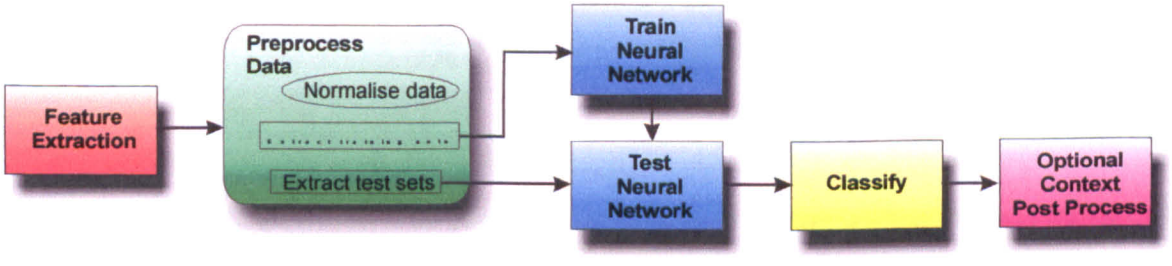


Fig.4.34 Neural Network based pattern recognition system

Such a system includes: a feature extraction unit, a module for preprocessing the raw data, a module for training and testing the NN architecture. The preprocessing consists of normalising data and extracting training and test sets from the experimental data (the feature data). Additionally, some statistical computations can be performed on the data obtained by the feature extraction process.

The training sets are then used to train the neural network architecture with the chosen learning algorithm. The NN architecture is then tested on the test sets. If the test is not satisfactory, the learning (training) continues until the desired performance is obtained. The performance is usually measured in percentage of recognised unseen patterns (true positives). This is called the prediction rate. It is usually a trade-off between the number of false rejections (false positives) and the number of recognised patterns (true positives). In other words, the more patterns the NN recognises, the more false positives are introduced in the classifying process. The designer should very carefully decide the optimal values for the system to be designed, in order to obtain an acceptable overall performance.

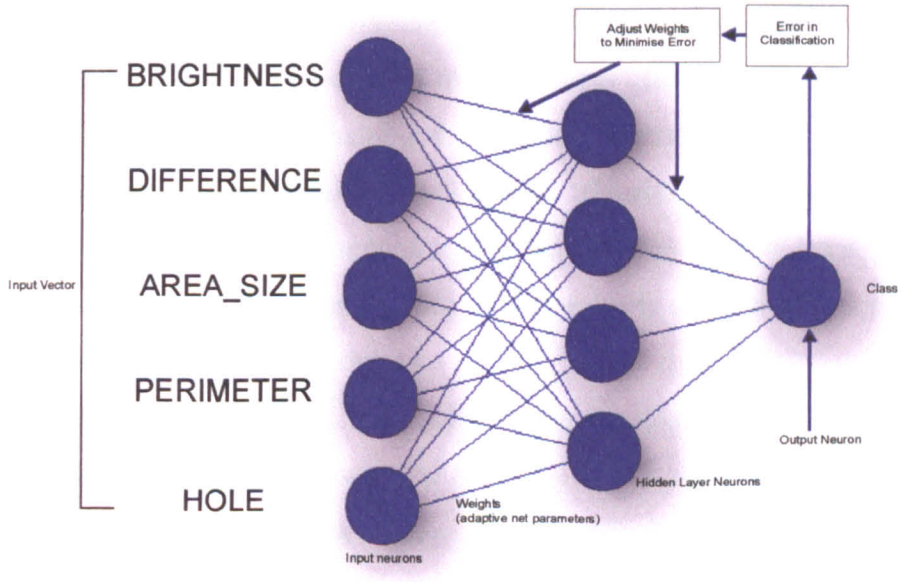


Fig.4.35 MLP architecture

As mentioned before, from the feature extraction process, five different features were gathered for an object that has to be classified as being a foreign body or not. A single MLP architecture (MLP) can be trained using those sets of data. Thus, the MLP will have five input nodes for the input layer – BRIGHTNESS, DIFFERENCE,

AREA_SIZE, PERIMETER and HOLE; a hidden layer and an output layer with only one node (see Figure 4.35). The output node will have binary output as follows:

- 0 – the area in question is not a foreign body (normal meat area)
- 1 – the area in question is a foreign body

A classic Back Propagation learning algorithm (BPNN) was designed and implemented.

4.5.2.2.1 BPNN algorithm

Neural network classifiers imply two fundamental aspects: learning and knowledge representation. Neural network approach of those aspects is different from the traditional symbolic approach. A neural computation is characterised by the possibility of learning based on partial information or in the presence of noise (erroneous data). Neural networks gather knowledge through learning or training. A neural network is considered trained when if applying input vectors of data produces at the output level expected outputs or at least consistent outputs. The knowledge gained through this process is stored by the weights of connections between neurons.

The training vectors (or training patterns) are presented to the network sequentially, and the weights are adjusted to "remember" the knowledge represented by those vectors. The process of adjusting the weights is made accordingly with a pre-established procedure called training or learning algorithm. Generally, every input vector from the training set is presented to the network more than once. The training algorithm has to determine the convergence of the weights to such values so that every input vector produces the expected output.

There are two fundamental types of learning associated with a NN: supervised and unsupervised learning *[Dumitrescu and Hariton, 1996]*.

In the case of supervised learning, the network is presented with a series of training sets or examples. A training set is a pair formed by an input vector and the corresponding desired output. When the input vector is applied to the neural network, the network output is computed and compared with the expected or wanted output. The difference between the expected output and the actual output is called the error. The weights are then adjusted according to an algorithm in order to minimise the error. The input vectors are sequentially and cyclically applied, adjusting the weights until the global error associated with the training set reaches an acceptable value *[Zurada, 1992]*.

In some models, the weights are adjusted only when the network output values is different from the expected values. This is the case of learning by error correction. In others, any input vector causes a weight adjusting process.

A major problem in AI is establishing a method of storing everything that has been taught. In a connectionist, knowledge representation is a network of weights associated with the neurons. Other important aspect is that when the neurons compute their activation's, other neuron activations are taken into account.

In this NN-based PR approach, the learning algorithm chosen is Back-propagation. It is a form of supervised learning.

To begin the discussion of training the network, it must first be recognised the need for a measure of how close the network has come to an established desired value. This measure is the network error. Therefore, first an error measure must be defined. Typically for the back-propagation *[Rummelhart et al., 1986]* training algorithm, an error measure known as the mean square error is used. The mean square error is defined as follows:

$$Err = \frac{1}{2}\sum_{j=1}^{N}\left(expected\_output_j - actual\_output_j\right)^2 \tag{4.52}$$

where N is the number of output neurons.

Back-propagation is one of the simplest members of a family of training algorithms collectively termed gradient descent. The idea is to minimise the network total error by adjusting the weights accordingly. Gradient descent provides a means of doing this. Each weight can be viewed as a dimension in an N-dimensional error space. In error space, the weights act as independent variables and the shape of the corresponding error surface is determined by the error function in combination with the training set. The negative gradient of the error function with respect to the weights then points in the direction that will most quickly reduce the error function. If one moves along this vector in weight space, one will ultimately reach a minimum at which the gradient becomes zero.

The back-propagation learning algorithm comprises two main phases:

- A direct forward pass through the network, from input layer to the output layer, in which the network is activated and the output values are determined;
- A backward pass through the network, from the output layer to the input layer, in which the real outputs are compared with the expected outputs for each input patterns and an error estimation is computed; this error estimation is then back propagated and utilised for adjusting the weights accordingly.

To apply the back-propagation algorithm the network weights must first be initialised to small random values. It is important to make the initial weights "small". Choosing initial weights too large will make the network untrainable. After initialisation, training set vectors are then applied to the network. Running the network forward will yield a set of actual values. Back-propagation can then be utilised to establish a new set of weights. The total error should decrease over the course of many such iterations. If it does not, an adjustment to the training parameters may be required.

One full presentation of all the vectors in the training set is termed as an epoch or cycle. When the weights approach values such that the total network error, over a full epoch, falls below a pre-established threshold the network is said to have converged. The error does not fall necessarily uniformly. Local fluctuations in the total network error are normal and expected, especially early in the training cycle.

The architecture used has the following parameters (see Figure 4.35):

- number of input nodes : 5
- number of hidden nodes: 4
- number of output nodes: 1

The following notations used are:

- w1[] - array of weights from input to hidden layer
- w2[] - array of weights from hidden to output layer
- x[] - array of input values
- h[] - array of hidden nodes outputs
- y - output of neural networks
- η - learning rate parameter
- output_error - the error at the output layer level

- hidden_error - the error at the hidden layer level

The stopping condition can either be when the worst error for every epoch (cycle) is lower than a specified threshold (network converged), or when the maximum number of epochs are reached (network failed to converge).

A customised version of back-propagation learning algorithm according to conditions under investigations is presented below:

BPNN algorithm

Step 1. Random initialisation of weights

w1[i,j]=random value within [-1..1] interval of real values; i=1-5; j=1-4; (4.53)
w2[i]=random value within [-1..1] interval of real values; i=1-4

Step 2. For every input pattern

2.1 Compute the network output and the hidden node outputs (assuming that each neuron in the network operates by taking the sum of its weighted inputs and passing the result through a non-linear activation function, in this case the sigmoid):

$$h[j]_{j=1,4} = \frac{1}{1+e^{-\sum_{i=1}^{3} w1[i,j]X[i]}} \qquad \text{j=1-4} \tag{4.54}$$

$$Y = \frac{1}{1+e^{-\sum_{i=1}^{4} w2[i]h[i]}} \tag{4.55}$$

2.2 Compute the errors at the output layer level

$$output\_error = Y \cdot (1-Y) \cdot (\exp ected\_output - Y) \tag{4.56}$$

2.3 Compute the errors at the hidden layer level

$$hidden\_error[j] = h[j] \cdot (1-h[j]) \cdot output\_error \cdot w2[j] \qquad \text{j=1,4} \tag{4.57}$$

2.4 Update weights

$$w2[i]_{t+1} = w2[i]_t + \eta \cdot output\_error \cdot h[i] \qquad \text{i=1-4} \tag{4.58}$$

$$w1[i,j]_{t+1} = w1[i,j]_t + \eta \cdot hidden\_error[j] \cdot h[j] \qquad \text{i=1-5; j=1-4} \tag{4.59}$$

Step 3. Repeat Step 2 until stopping condition reached

4.5.2.2.2 Experimental investigations

Usually, the performance of a NN-based system is determined as being the percentage of correct recognized input patterns. However, a NN's performance is influenced by the architecture parameters such as the network error and the learning rate

value. Investigations were performed in order to select optimal parameters for the NN-based PR architecture.

Choosing an adequate value for the learning rate parameter η has a significant effect over the neural network performance. Usually, the value should be within [0,1] interval of real numbers, but it is possible that this interval could be too large. The gradient of the error function has to converge to zero, thus the algorithm needs to find the global minimum of the error function. Regularly, choosing a small value for the learning rate, i.e. within [0.05, 0.25] of real values, a solution is more likely to be obtained even though the neural network will need a large number of training cycles (epochs). It is also possible to raise the value of this parameter during the training, gradually, as the global error decreases. This will generally lead to increases for the speed of learning. In this case, there is the risk that the network may "jump" too far from the real minimum value for a too large η. The increase of η during training is useful only if the error surface is relatively plain. In this case, as the network is getting close to a solution, the derivative of the error becomes smaller all the time, which leads to non-significant changes to the weights *[Toderean et al., 1994]*. Generally, trying to raise the value of a learning rate can be justified in order to speed up the training process. However, the speed of the training process is not very important since this process of training is performed only once. Thus, this process of using an adaptive learning rate will not offer an improvement to the problem solution. Furthermore, it may influence the performance of the neural network in a negative way such that some global minima may be skipped and the network will reach only a local minimum and the network will start to oscillate.

The problem here is to find an optimum value for the learning rate parameter for the present problem. The value is problem-specific, depending on the error surface. When η is chosen too small, the gradient is closely followed and the process of convergence to a solution is relatively slow. When η is chosen too large, the process of learning is faster but there is the possibility of oscillation. In the end, choosing a value for this parameter is a matter of experimentation.

The values chosen for the learning rate and the performance obtained are presented in the following table. As it can be seen, the optimum values for the learning rate is 0.1.

When measuring network testing performance to a test vector the probability with which the NN gives a result is defined as a test error. In other words, if the test error is 0.1, the NN gives an answer with a 0.9 degree of truth (90% probability). The performance of the NN classifier for different test errors is presented in Table 4.10. It is obvious that when the test error is small, the probability that the NN gives a correct answer increases, but at the same time the performance decreases. In contrast, a high value for the test error will render a better network performance, but the accuracy of the output will decrease accordingly. In general, the test error is chosen to be of a similar value with the training error. Since the training value is 0.1, the test error was chosen to have the same value. The performance of the classifier is in this case over 90% (as highlighted in Table 4.11).

Table 4.10 Choosing the learning parameter for MLP architecture

Learning rate	Total number of patterns	Correct classified patterns
0.7	628	549
0.5	628	556
0.35	628	563
0.2	628	564
0.1	628	565
0.05	628	562

Table 4.11 MLP performance

Test error	Total no. of patterns	Correct classified patterns	Performance
0.05	157	140	89.17%
0.1	157	142	90.44%
0.15	157	143	91.08%
0.2	157	143	91.08%

4.5.2.2.3 Alternative NN approach

In order to meet the flexibility issues presented in paragraph 4.5.2, a different architecture was implemented, this being a Competitive Multi Modules NN (CMMNN). It is a modification of author's previous work *[Amza, 1999], [Amza et al., 2000]*. It consists of an assembly of BPNNs, as depicted in Figure 4.36.

Table 4.6 was used in order to address flexibility issues as an end-user option. The idea behind CMMNN lies in *[McKenna et al., 1992]*. Their approach consists of a two-stage architecture. The two stages were independent in the sense that first stage

completes its learning phase prior to commencement of stage two learning. It is actually a type of decomposition the original problem in the traditional manner, following the old roman principle "divide et impera" (divide and conquer). Thus, specialised modules are introduced for different types of foreign bodies. A voter has to make the final decision. Due to graceful degradation in the presence of noise, stage 1 errors will be minimised or reduced to nothing by stage 2 of this architecture, improving the global performance of the pattern recognition system.

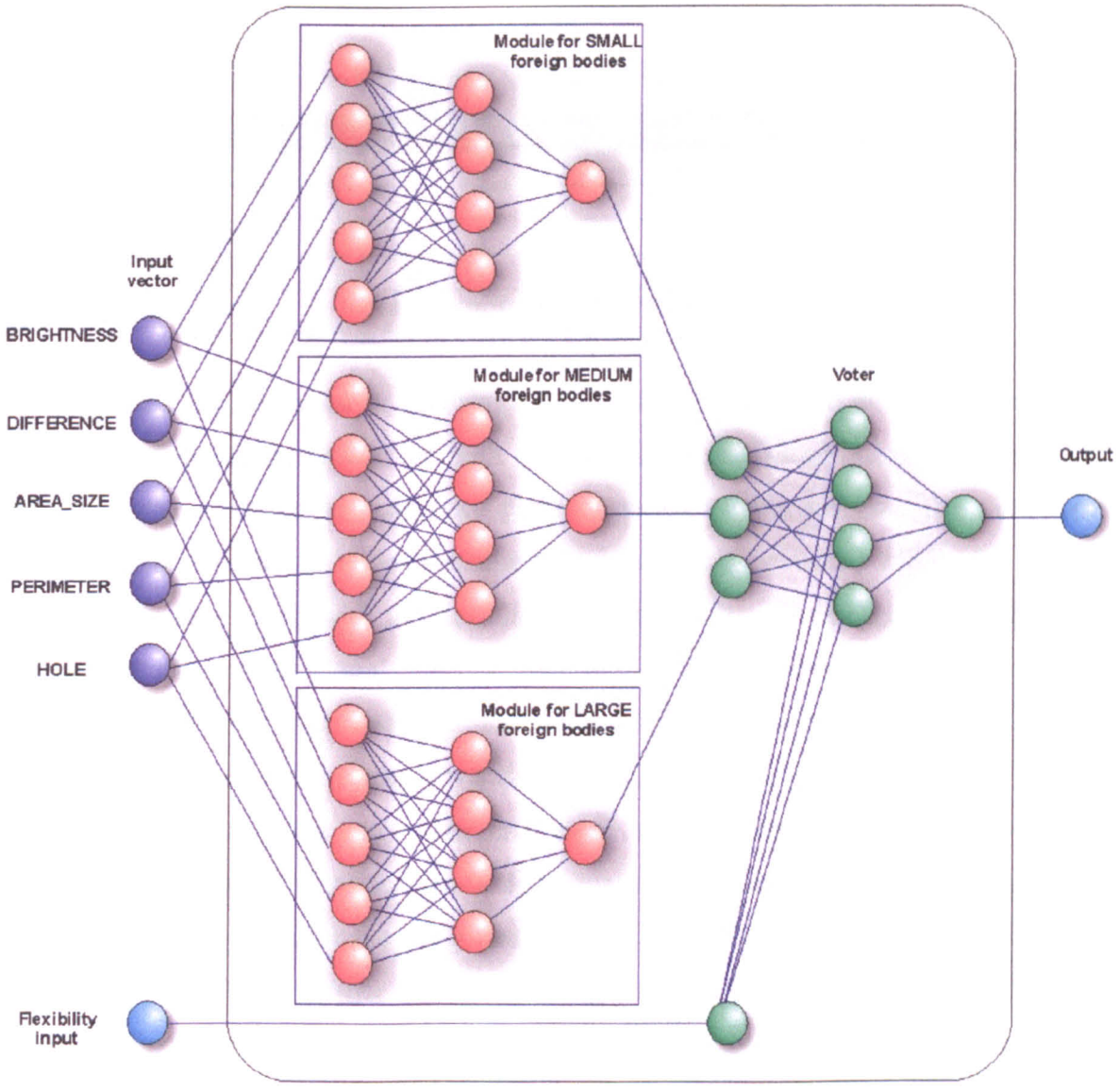


Fig.4.36 Competitive Multi Module Neural Network

Thus, CMMNN architecture will have two independent stages. The first stage comprises foreign-body specialised modules and the second stage comprises the specialised flexibility module or voter. Each module in the first stage is trained to

correctly recognize a specific type of foreign body. The voter is trained with specific data vector strings created according to the number of flexibility classes defined. Its training process is highly dependant on a parameter called the *flexibility input* that can be set by the final user. In the present case, for four classes of flexibility, the uniformly chosen values for the flexibility input (over the [0,1] interval of real numbers) are presented in Table 4.12, along with their corresponding expected inputs from the foreign-bodies specialised modules. Training data was generated from Table 4.5 (32 cases).

Table 4.12 Flexibility input node and associated flexibility classes

No	FI	SMALL	MEDIUM	LARGE	Description of flexibility class
1	0	1	1	1	ALL=SMALL+MEDIUM+LARGE
2	0.25	1	0	0	SMALL
3	0.50	0	1	0	MEDIUM
4	0.75	0	0	1	LARGE

1- this category of foreign bodies has to be rejected;
0 – this category of foreign bodies is not important for the process of rejection

Every module consists of a single MLP module, as shown in Figure 4.36. After training all weights are frozen for both stages and the prediction phase can commence. The two stages of learning can take place in parallel. After training, the same input vector will be fed into all bone-specialised modules and their results will be fed into the flexibility module along with the flexibility parameter. For consistence and ease of use the output of the first stage modules were named as the represented classes' name: SMALL, MEDIUM and LARGE. The output of the system is the output of the flexibility module with the same convention for the above MLP.

The output at the end of the final stage is computed and a WTA algorithm is applied on the output of the foreign-body specialised modules. The reason behind this is to facilitate recognition for the voter that expects, according to its training, one input of a high value while the other inputs are insignificant (since a foreign-body can only belong to one category, or belong to one category much more strongly than to another). In essence, the voter chooses the highest output from the first stage output and it propagates it to the final output only if the flexibility input allows it. However, when the end-user is interested in detecting all foreign-bodies, it does not matter what specialised foreign-body module has the highest output.

As mentioned previously, the CMMNN training performance depends heavily on the parameters value. Different performance obtained with different learning rates and different number of hidden nodes for the hidden layer are presented in Table 4.13. From the computational overhead point of view the chosen architecture was one with only 4 hidden nodes per module, an architecture that at a learning rate of 0.01 reaches the same performance as one with 5 hidden nodes at a 0.35 learning rate (as highlighted in Table 4.13).

Table 4.13 Choosing the learning parameter for the CMMNN voter architecture

Learning rate	No. of hidden nodes	Total number of training patterns	Number of learned patterns	Rate of performance for training sets
0.5	4	32	24	75%
0.35	4	32	24	75%
0.2	4	32	25	78.12%
0.1	4	32	26	81.25%
0.01	4	32	30	93.75%
0.5	5	32	26	81.25%
0.35	5	32	30	93.75%
0.2	5	32	28	87.5%
0.1	5	32	29	90.62%
0.01	5	32	27	92.19%
0.5	8	32	28	87.5%
0.35	8	32	26	81.25%
0.2	8	32	28	87.5%
0.1	8	32	27	92.19%
0.01	8	32	29	90.62%

The testing performance of the CMMNN is presented in Table 4.14.

Table 4.14 CMMNN global testing performance

Flexibility category	Test error	Total no. of patterns	Correct classified patterns	Performance
ALL	0.1	157	142	90.44%
SMALL	0.1	35	31	88.57%
MEDIUM	0.1	98	88	89.79%
LARGE	0.1	24	20	83.33%

4.5.2.2.3 Noise concerns

As far as the noise presence in the input vectors is concerned, the results for MLP are presented in Table 4.15 and for CMMNN in Table 4.16. The change in

performance when the noise level increases is not as steep as the change in the classical classifier (see Figures 4.37 and 4.38). Furthermore, due to the graceful degradation in the presence of noise property of an NN both architectures perform very well, even when the noise level is as high as 15%.

Table 4.15 Noise level and the corresponding performance of the MLP

Noise level	3%	5%	10%	15%	20%	25%	30%
Performance	89.3%	88.1%	85.3%	81.2%	73%	68.6%	57.5%

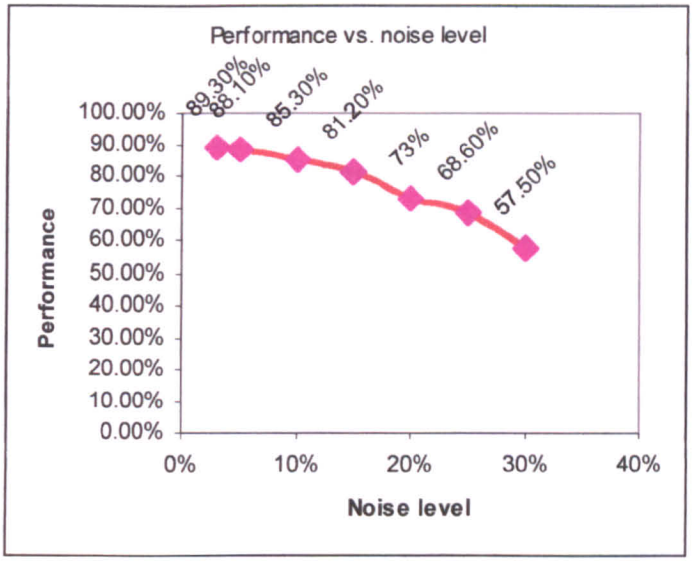


Fig.4.37 Performance vs. noise level for MLP

Table 4.16 Noise level and the corresponding performance of the CMMNN

Noise level	3%	5%	10%	15%	20%	25%	30%
ALL	89.2%	87%	84.2%	81.3%	75%	67.8%	58.2%
SMALL	87%	82.3%	78.8%	73.2%	70.9%	65.1%	53.2%
MEDIUM	87.1%	80.9%	73.2%	68.7%	61.3%	58.9%	48.2%
LARGE	81.1%	77.3%	72.4%	64.2%	59.7%	53.2%	49.8%

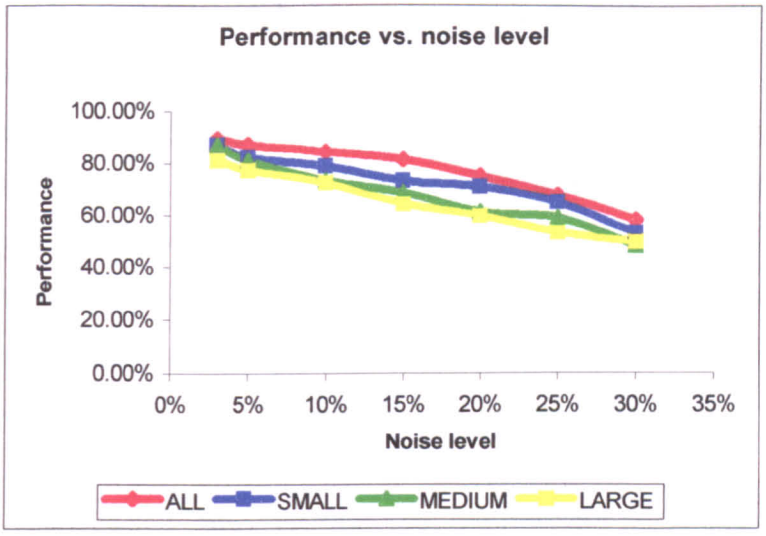


Fig.4.38 Noise influence over CMMNN performance

The performance obtained by the three presented approaches so far stays below 90%. However, the architectures lack flexibility from the input vectors point of view. Hard decisions are taken in the classical approach and there is no room for errors or small differences between input vectors. Small differences in the characteristics of an object are translated into small differences for the input vector associated with that object. The classical approach does not deal well with changes in the input patterns, since a hard comparison is made using the already stored and (unchanged over time) patterns in the look-up table. The NN approach offers the end user a little more flexibility, but the results are not as good as expected. Changes in the input patterns affect the NN's performance and as do the stored patterns within the network connections. NNs can deal with small changes in the input vectors, due to their property of graceful degradation in the presence of noise, but the architecture is not flexible enough. Changes can appear in the features extracted from objects for a series of reasons: different illuminations of a product, change over time in X-ray source properties, different thickness of product, different types of meat, etc. This leads to a need to systematically deal with uncertainty. Therefore, a fuzzy logic approach was implemented as a PR system.

4.5.2.2.4 False positives and false negatives

As defined before, a false positive is when the network architecture is rejecting an input pattern that should not be rejected. If an input pattern is classified as a foreign

body when in fact it is not one, a false positive appears. The number of false positives is very important in the food industry. A good and 'clean of foreign bodies' product being rejected will lead to increases in production costs and time by sending that product to be processed again. Thus, this number of false positives needs to be minimised. Also, no food factory wants to miss a large number of foreign bodies, because this will affect sales and most probably relationships with its customers. Misses are called false negatives and are defined as foreign bodies that are reported as being normal meat, therefore not being rejected. The number of false negatives also needs to be kept to a minimum.

In the present case, a false positives rate (or fraction) was computed as follows: the number of non-foreign body input vectors reported as being foreign bodies divided by the number of total normal meat input vectors (in percentage). In computing the number of normal meat patterns reported as foreign bodies a false positive error was introduced. This is the probability that non-bone data is reported as bone data, i.e. a false positive error of 0.1 means that if the network response to a non-bone input pattern is at least 0.9 this is considered as being classified as bone with 90% probability.

The results for both MLP and CMMNN architectures are shown in Table 4.17 and 4.18.

Table 4.17 False positives fractions for MLP

Testing error	False positive error	False negatives rate	False positives rate
0.1	0.01	0%	0%
0.1	0.1	0.45%	0.69%
0.1	0.5	6.82%	2.31%
0.1	0.8	11.36%	4.40%
0.2	0.01	0%	0%

It is obvious from how the false positive error was defined that for a large value of the false positive error the false positive rate increases considerably. As long as the false positive error is kept under a minimal threshold the architectures perform well and the number of false positives will be relatively low (no more than 7.5%). False negatives are not influenced by this error and they vary in range from 1% to 5%, values that can be considered quite satisfactory.

Table 4.18 False positive fraction and false negative fraction for CMMNN

Flexibility category	Testing error	False positive error	False negatives rate	False positives rate
ALL	0.1	0.01	1.36%	7.64%
	0.1	0.1	1.82%	7.64%
	0.1	0.8	2.27%	99.31%
SMALL	0.1	0.01	4.09%	2.08%
	0.1	0.1	4.09%	2.55%
	0.1	0.8	5.00%	2.78%
MEDIUM	0.1	0.01	0%	0.69%
	0.1	0.1	0%	0.93%
	0.1	0.8	0%	0.93%
LARGE	0.1	0.01	3.64%	0.46%
	0.1	0.1	4.09%	0.46%
	0.1	0.8	4.55%	0.69%
	0.1	0.8	2.73%	0.46%

4.5.2.3 Fuzzy approach

Fuzzy logic (FL) is an easy and convenient approach for mapping an input space (referred as a universe of discourse) to an output space *[Kosko, 1992],[Cox,1994]*. In other words, mapping inputs to their corresponding outputs, such as in a general PR system where the inputs can be the patterns and outputs are the classification classes. Applying a FL approach to a PR system brings FL properties such as:

- great flexibility, due to the ease of modifying or adding more functionality to the system;
- great tolerance of imprecise data
- experience of human experts can be use as a scaffolding for FL;
- FL is based on natural language; natural language has evolved over the centuries to be convenient, easy and efficient, therefore it is safe to assume that FL is also easy to use when compared to other methods such as NNs.

FL is based on common sense statements. Natural language that has meaning both to the human user and the machine is used by FL as a technique to bridge the gap between people and machines.

The main concept of FL is the *fuzzy set*. A *fuzzy set* is a set without a crisp, clearly defined boundary. As opposed to a classical set that wholly contains or wholly excludes any given element, a fuzzy set can contain elements with only a partial degree of membership. This *degree of membership* is also known as the *membership function* (MF) since it establishes a correspondence between an element in the domain and a truth value indicating its degree of membership in the set. A membership function is a curve that defines how each point in the input space is mapped to a membership value (between 0 and 1).

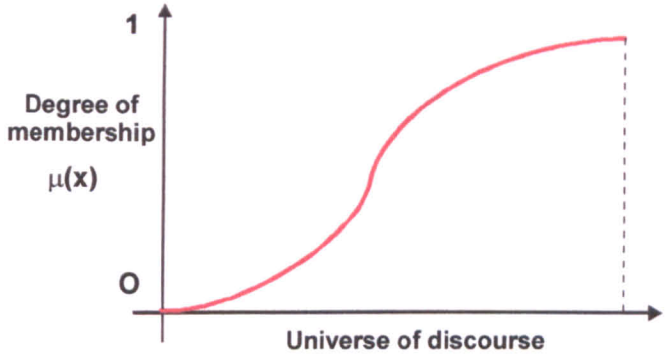


Fig.4.39 General fuzzy set representation over a universe of discourse

A general description of a fuzzy set A is presented in Figure 4.39. A fuzzy set can be defined as follows:

$$A = \{x, \mu_A(x) \mid x \in X\} \tag{4.60}$$

where X is the universe of discourse, x are its elements and $\mu_A(x)$ is called the MF of x in A. Typical fuzzy sets are presented in Figure 4.40.

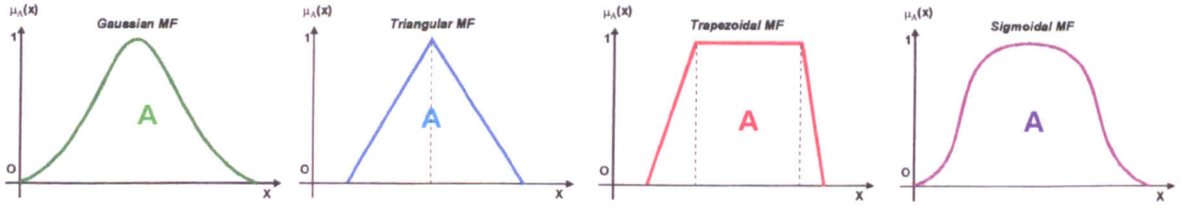


Fig.4.40 Typical fuzzy sets

4.5.2.3.1 General fuzzy inference system

A FL-based PR system for the food inspection process is presented in Figure 4.41. As with all implementations of PR systems, it must contain an object and feature extraction unit and a classifier. The classifier is a FL-based one called *fuzzy inference system*. This classifies the current object as a foreign body or normal meat.

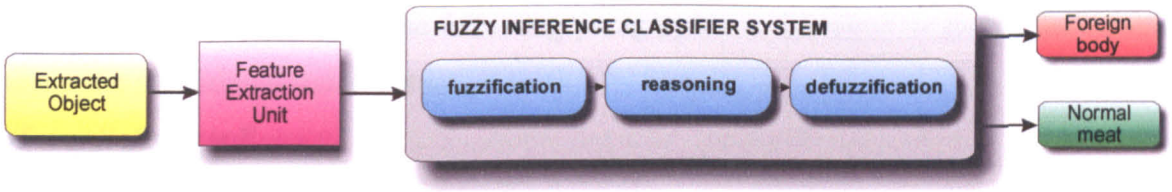


Fig.4.41 General FL-based PR system for the food inspection process

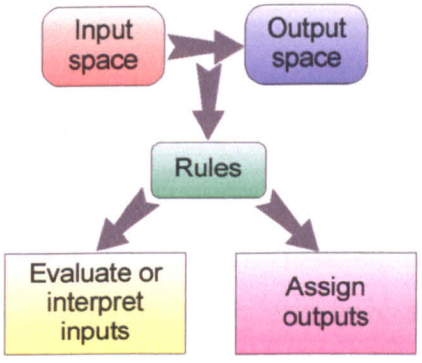


Fig.4.42 General input-output space mapping process

In general, fuzzy models manipulate *linguistic variables*. A *linguistic variable* is the representation of a fuzzy space derived from the evaluation of the linguistic variable. The simplest linguistic variable is the name of the fuzzy set directly representing a specific area in the problem space. Furthermore, fuzzy models consist of a series of conditional or unconditional *fuzzy propositions* or *statements* or *rules*. Thus, the mapping process between the input space and the output space is just a matter of evaluating the fuzzy linguistic-based rules and assign outputs according to their results (see Figure 4.42).

A proposition creates a link between a value in the fuzzy domain and a fuzzy space. A general form of a proposition is:

$$x \text{ is } Y \tag{4.61}$$

where x is a fuzzy variable (scalar in the domain space) and Y is a linguistic variable. The process of evaluating a fuzzy statement is the grade of membership derived from the transfer function:

$$\mu \leftarrow (x \in Y) \tag{4.62}$$

In other words, a proposition establishes to what degree is x a member of set Y. A specific case of propositions is the conditional case. A conditional proposition

(referred to as *fuzzy rule* in the following paragraphs) is qualified by an *if*. The general form of a conditional statement is:

if a is B then x is Y (4.63)

where a and x are scalar values and B and Y are linguistic variables. The proposition or propositions in between *if* and *then* are the *antecedent* or *predicate* of a fuzzy conditional proposition. The proposition following the then term is the *consequent* .

All fuzzy rules of such a system are evaluated in parallel and the order of rules is not important. Evaluation of a fuzzy rule assumes two steps:

1. evaluating the antecendent; that implies the fuzzification of inputs process that resolves all the inputs to a degree of membership between 0 and 1; if there are multiple parts of the antecedent, apply necessary fuzzy operations and resolve the antecedent to a single number between 0 and 1; and
2. apply the result to the consequent, a process known as implication; the consequent of a fuzzy rule assigns an entire fuzzy set to an output.

A general FL system comprises more than one rule. All rules render a fuzzy set. The fuzzy sets are then aggregated into a single fuzzy set. Using the general rules of fuzzy inference the evaluation of a proposition produces one fuzzy set associated with each model solution variable. To find the actual value of the corresponding output scalar of a FL system, a process called defuzzification needs to be performed. Defuzzification is the final stage of fuzzy reasoning. It is the process of resolving the output fuzzy set to a single number.

The process of mapping inputs to the desired outputs by mean of FL is also called fuzzy inference. From the way the outputs are computed, there are two types of fuzzy inference systems:

- Mamdani type systems – expect the output membership functions to be fuzzy sets
- Sugeno type system - any inference system in which the output membership functions are either linear or constant.

4.5.2.3.2 A proposed fuzzy inference system

The proposed architecture consists of two fuzzy filtering modules and two feature extraction units (Figure 4.43). The first feature extraction unit extracts the

geometrical characteristics of the current object. Using these features, the fuzzy geometrical pre-filtering unit computes the probability of being a foreign-body for the currently analysed object (PROB). Then, the second features extraction unit computes the object's grey-level based properties. Based on the probability output (PROB) of the first fuzzy module, the fuzzy grey-level based statistical filtering module takes the final decision about the current analysed object.

Time is of essence in any real-time application and the present food inspection implementation is no exception. Therefore a second fuzzy approach was implemented. The design is similar to the first approach above. Two feature extraction units and two fuzzy inference modules are still present in the proposed architecture (see Figure 4.44). The first module decides whether an object or area corresponds as being a foreign body from the geometrical point of view. However, as opposed to the first case, its output is not inputted in the second fuzzy module. It is analysed and a decision is taken whether the system has to continue the analysis process. Thus, the second module is used only if the output of the first module confirms the possibility of the current area being a foreign body. It consists of a second fuzzy inference system using statistical grey-level characteristics of the region in question. If the object corresponds to the statistical grey-level foreign body criteria, the food product is rejected. Otherwise, the next object is analysed and the process continues until all objects are analysed or one object is found to be a foreign body. However, the process of detection of foreign bodies carries on no matter how many are found. The reason behind this is so that the performance of this approach can be measured.

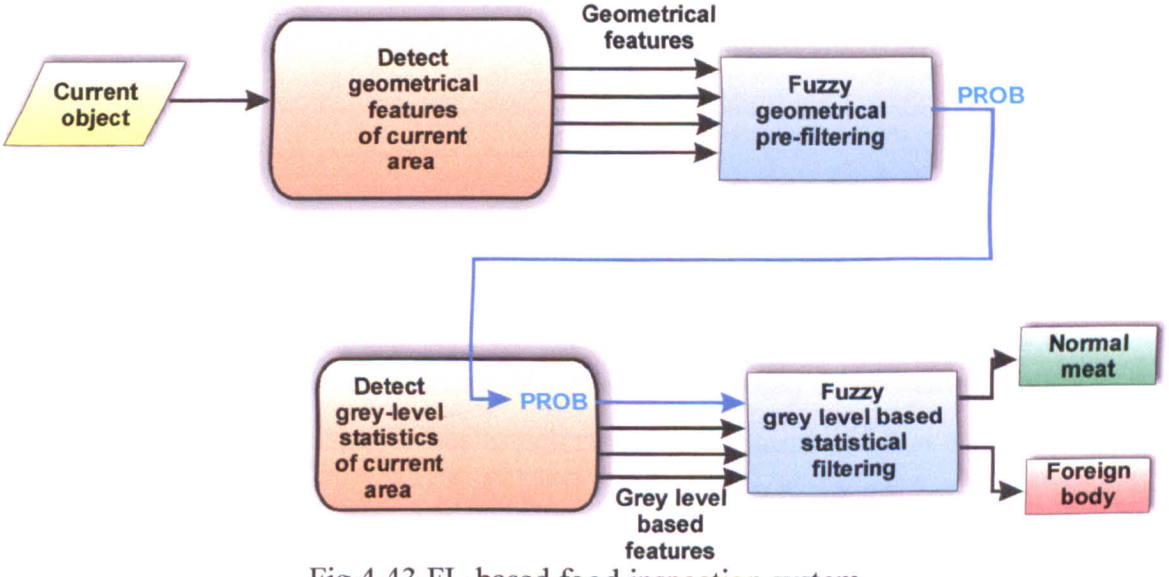


Fig.4.43 FL-based food inspection system

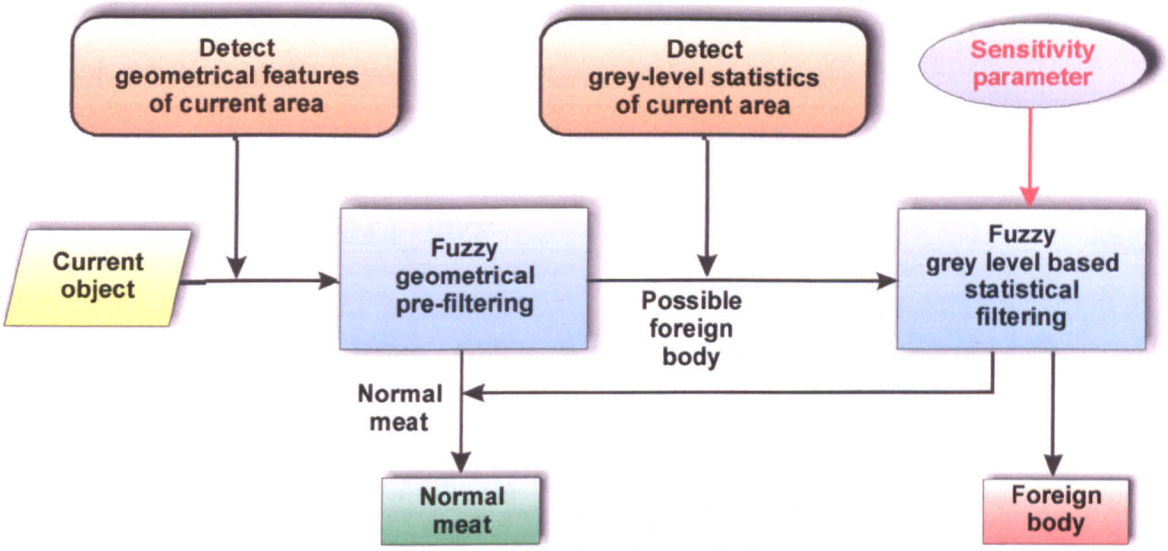


Fig.4.44 Proposed fuzzy architecture

4.5.2.3.3 Fuzzy geometrical pre-filtering

4.5.2.3.3.1 Fuzzy measures definition

A first step was to define the standard fuzzy sets describing the geometrical properties of the generated areas. From each region, five features are extracted. The first and the most important is the number of pixels contained in that area: **AREA_SIZE**. The second measure is the size of the air holes contained within the object: **HOLE**. This reflects the homogeneity of the area in question. These measures are defined as fuzzy sets using trapezoidal membership functions. **PERIMETER** is another important computed measure. It consists of the number of pixels at the edge of the region in question. The **SHAPE** feature is a measure of the shape of an individual region. The compactness of a shape is derived from the ratio of the perimeter of the shape divided by its area:

$$SHAPE = \frac{PERIMETER}{AREA} \tag{4.64}$$

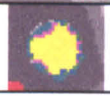
The **ROUNDNESS** measure *[Hiltner et al., 2001]* is defined as follows: if a region is a circular one then it has a high roundness measure; if it is an ellipse or of a different shape then it has a low roundness measure. In defining this concept a reference circle was used with its centre in the centre of gravity of the current region and with a similar area size. Therefore the concept was defined as the difference between the two

areas (pixels that do not belong to the region but belong to the reference circle-A_1; and pixels that belong to the region but do not belong to the reference circle-A_2):

$$ROUNDNESS = 1 - \frac{A_1 + A_2}{2 \cdot AREA\_SIZE} \tag{4.65}$$

Examples for calculation of different measure are presented in Table 4.19.

Table 4.19 Roundness measure

Roundness	Area size	Image
31.18%	1076 pixels	
64.03%	958 pixels	
72.80%	410 pixels	
88.54%	240 pixels	

The linguistic variables defined for the measure are presented in Table 4.20. The crossover points between neighbouring membership functions were determined experimentally. Their corresponding fuzzy sets are also presented in Figure 4.45.

Table 4.20 Linguistic variables defined for both fuzzy filtering modules

Geometrical features
AREA_SIZE: tiny, small, medium, big, xlarge
HOLE: small, big
PERIMETER: short, long
SHAPE: sinuous, compact
ROUNDNESS: low, medium, high

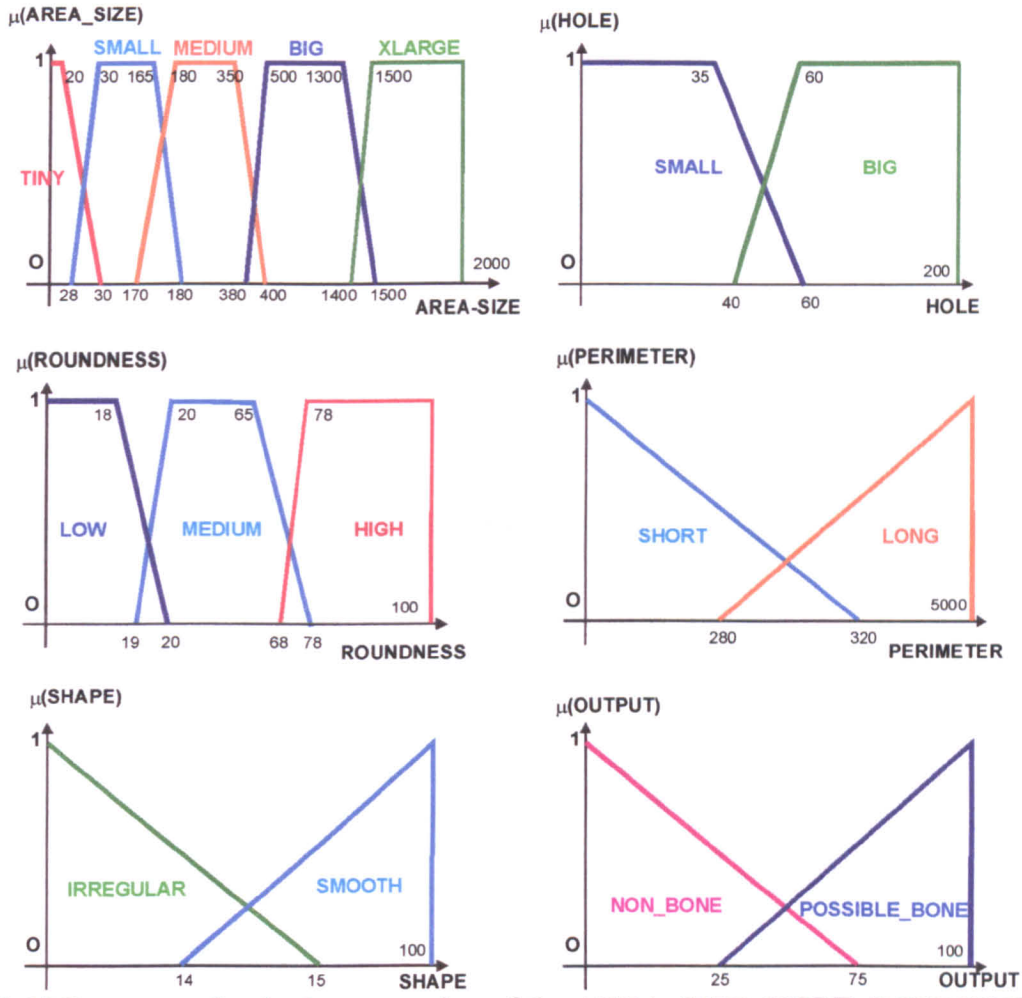


Fig.4.45 Fuzzy sets for the interpretation of the AREA_SIZE, HOLE, ROUNDNESS, PERIMETER, SHAPE and OUTPUT concepts

4.5.2.3.3.2 Fuzzy pre-filtering inference system

A Mamdani-type fuzzy inference system with 5 inputs, a single output and 11 rules that apply geometrical constraints was built *[Berks and Keyserlingk, 1998], [Zimmerman, 1996].* The output of the fuzzy system is a fuzzy set. This fuzzy set needs to be defuzzified in order to determine the system decision as a single, crisp number. A classical approach was taken in the present case for this process. The centroid of the area of the output fuzzy set was computed as the crisp response value of the system. If the computed value of the output was below a certain threshold, then the response of the system was considered negative (NON_BONE), otherwise the system's response was considered positive (POSSIBLE_BONE). Same mechanism was used for the second fuzzy module.

The rules used are presented in Table 4.21. The basic principle for the area in question was determined experimentally. Human perception was used to define the

object criteria of being a foreign body or not. If the area is within its accepted size and it does not contain big holes and it has an accepted measures of shape and roundness, then the object is treated as being a possible bone; otherwise, the object is considered as being non-bone and it is eliminated from later high-level detection (second fuzzy grey-level statistics module is not necessary). Rule acquisition based on human perception is an advantage over the NN approaches, where the data is simply extracted from the objects, but no human knowledge is used when training the architectures.

Table 4.21 Rules for Fuzzy Module 1

NORMAL MEAT (NO_BONE) RULES
1. if AREA_SIZE is TINY then OUTPUT is NON_BONE 2. if AREA_SIZE is XLARGE then OUTPUT is NON_BONE 3. if HOLE is BIG then OUTPUT is NON_BONE 4. if SHAPE is IRREGULAR then OUTPUT is NON_BONE 5. if PERIMETER is LONG then OUTPUT is NON_BONE 6. if ROUNDNESS is LOW then OUTPUT is NON_BONE 7. if AREA_SIZE is MEDIUM and HOLE is SMALL and SHAPE is SMOOTH and PERIMETER is SHORT and ROUNDNESS is HIGH then OUTPUT is NON_BONE 8. if AREA_SIZE is BIG and HOLE is SMALL and SHAPE is SMOOTH and PERIMETER is SHORT and ROUNDNESS is HIGH then OUTPUT is NON_BONE
FOREIGN BODY (BONE) RULES
9. if AREA_SIZE is SMALL and HOLE is SMALL and SHAPE is SMOOTH and PERIMETER is SHORT then OUTPUT is POSSIBLE_BONE 10. if AREA_SIZE is MEDIUM and HOLE is SMALL and SHAPE is SMOOTH and PERIMETER is SHORT then OUTPUT is POSSIBLE_BONE 11. if AREA_SIZE is BIG and HOLE is SMALL and SHAPE is SMOOTH and PERIMETER is SHORT and ROUNDNESS is HIGH then OUTPUT is POSSIBLE_BONE

4.5.2.3.4 Fuzzy grey-level based statistical filtering

4.5.2.3.4.1 Fuzzy measures

The functionality of the second fuzzy module is similar to the geometrical fuzzy module. Its inputs consist of features extracted by the grey-level feature extraction unit. The features extracted are presented in Table 4.22. They are the BRIGHTNESS of the

area in question, the DIFFERENCE between area intensity and its neighbourhood intensity and the TOGETHERNESS of the region. The first measure corresponds to the mean intensity of the grey-level for the current region. The second measure is defined as the difference between the mean intensity for the region and the mean intensity for the neighbourhood of the same area. Both high- and low energy images were used when computing those two measures.

Table 4.22 Linguistic variables defined for second fuzzy module

Grey-level based features
BRIGHTNESS: low, high
DIFFERENCE: big, small, negative
TOGETHERNESS: loose, close

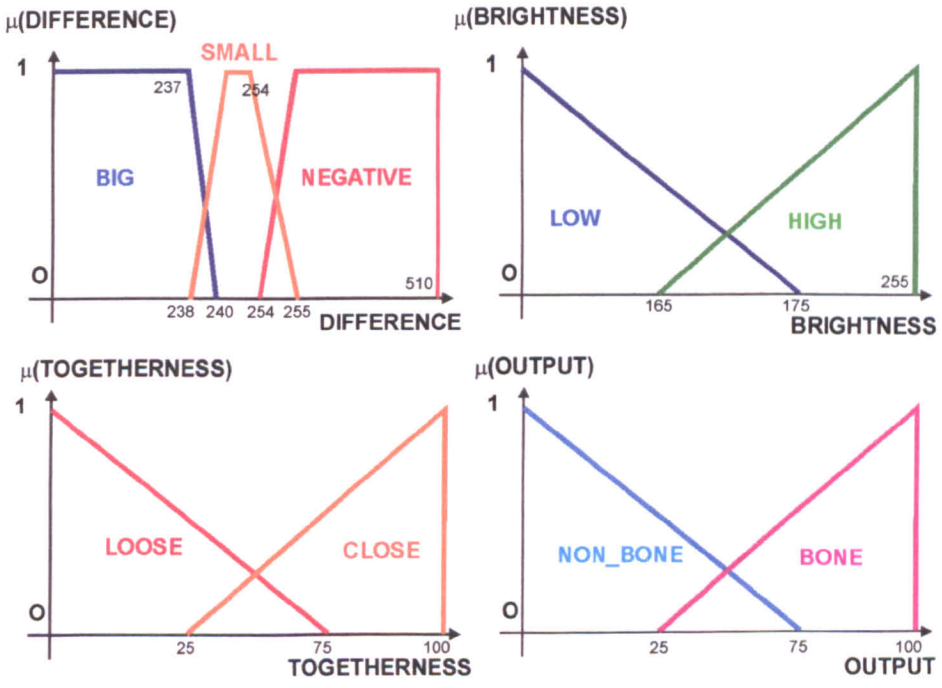


Fig.4.46 Fuzzy sets defined for the input measures of the second fuzzy module

The third measure refers to the homogeneity and heterogeneity of the region *[Jaulent et al., 1997]*. Small variations of grey level within the same region make it homogenous, whereas high transitions make it heterogeneous. The segmentation process should only provide the later stages of the detection process with homogenous objects. However, if the segmentation process provides heterogeneous areas, the TOGETHERNESS concept will be used as a means to reject those regions.

$$TOGETHERNESS = \sqrt{\frac{\sum_{pixel(i,j)\in Area}\left(I_{pixel(i,j)} - BRIGHTNESS\right)^2}{AREA\_SIZE}} \tag{4.66}$$

The fuzzy sets for the input measures are defined in Figure 4.46.

4.5.2.3.3.2 Fuzzy inference system

The second fuzzy-filtering module consists of 3 inputs and 5 rules, derived by summarizing the knowledge gathered from direct observations and human experts. A bright object that differs significantly from the background is considered a foreign body, whereas an object with a lower brightness than its surrounding background is treated as normal meat. If the object is not a homogenous area then it is considered as being normal meat. The rules are presented in Table 4.23.

Table 4.23 Rules for Fuzzy Module 2

NORMAL MEAT (NO_BONE) RULES
1. if DIFFERENCE is SMALL and BRIGHTNESS is LOW then OUTPUT is NON_BONE 2. if DIFFERENCE is NEGATIVE then OUTPUT is NON_BONE 3. if TOGETHERNESS is LOOSE then OUTPUT is NON_BONE
FOREIGN BODY (BONE) RULES
4. if DIFFERENCE is BIG and TOGETHERNESS is CLOSE then OUTPUT is BONE 5. if DIFFERENCE is SMALL and BRIGHTNESS is HIGH and TOGETHERNESS is CLOSE then OUTPUT is BONE

4.5.2.3.4 Experimental investigations

A crisp input was also added to the proposed fuzzy architecture as a method to set the sensitivity of the detection process. This was used in both fuzzy modules to decrease or increase the weight of specific rules according to the desired flexibility criteria (Table 4.24). Since the flexibility classes defined were ALL, SMALL, MEDIUM and LARGE, the easiest way to approach this is to decrease or increase the weight (or strength) associated with rules 9 and 10 of the first fuzzy inference system. If one sets the sensitivity parameter as MEDIUM, then the weight of rule 10 in the inference process is decreased to 0.1 so that it will not be taken into account by the

reasoning process. However, when the sensitivity parameter is set to ALL, a high sensitivity is set for the detection system (detection of all possible foreign bodies), the weight of all the rules remains 1 and they all are contributing to the final fuzzy aggregated output.

Table 4.24 Sensitivity parameter values and the corresponding rule weight adjustments

MODULE	Sensitivity parameter	ALL	SMALL	MEDIUM	LARGE
1	Rule's weight decreased	none	none	Rule 9	Rule 9 and 10
2	Rule's weight increased	Rule 5	none	none	none

In the second fuzzy module there are no rules regarding the size of a foreign body. However, rule 5 tells the system that even an area with a high brightness that is not very different from that of the surrounding background can be a foreign body. From practice, this is true in only a small percentage of cases. However, if one aims to detect all foreign-bodies, then that rule needs to be strengthened. Therefore, when the flexibility class is ALL the weight of rule 5 needs to be increased. In all other cases, the weight of rule 5 remains 0. Thus, rule 5 in the second fuzzy inference module is only used when the end-user wants to detect all possible foreign-bodies.

There is a trade-off between the sensitivity of the detection process and the false positives (area that are not bones but reported otherwise by the system). For a high sensitivity (ALL) the system will detect all of the foreign bodies, and may also detect generated false-positives. In contrast, for normal detection sensitivity (in any of the LARGE, MEDIUM or SMALL categories) no false positives are expected. However, from practice it is deduced that foreign bodies with a high brightness surrounded by a low contrast neighbourhood cannot be detected. One can set the sensitivity parameter in order to meets specific detection criteria.

4.5.2.3.5 Experimental investigations

A range of results is shown in Figure 4.47, for both bone and bone-free images.

As shown when a high sensitivity is employed the system renders more objects classified as foreign bodies (true positive along with false positive objects). As can be seen in general the results are quite satisfactory. However, false positives and false negatives may also occur (Figure4.47 f1-f4).

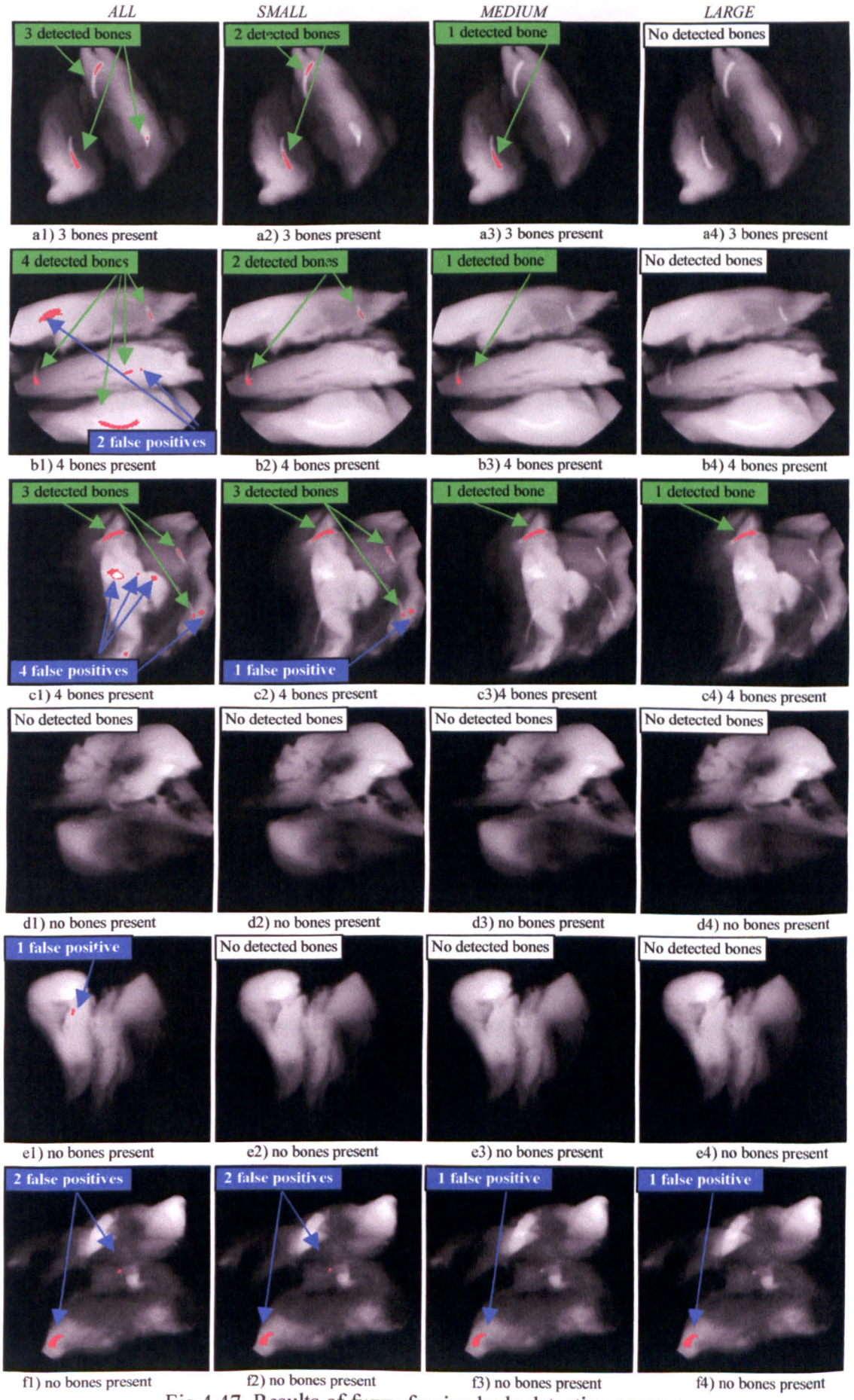


Fig.4.47 Results of fuzzy foreign body detection process

In Figure 4.47 b, one can see that there is a foreign body that it is not detected when the sensitivity is set to SMALL, MEDIUM or LARGE. However, combination between ALL sensitivity and one of the other three sensitivity classes SMALL, MEDIUM and LARGE are also possible. One can choose for instance to have a combination of ALL and LARGE. The system will then try to detect all large foreign bodies with high sensitivity but also more false positives may be introduced. The result when using (ALL, SMALL) class of sensitivity is depicted in Figure 4.48. An improvement is shown in the sense that another foreign body is detected successfully.

However, from a global point of view, when computing the performance accuracy of the system for all 4 classes of sensitivity, the results are excellent. A sample of 182 images (117 with foreign bodies and the rest without) was used in computing the accuracy of the system performance and the percentage of false positives and false negatives. The results are shown in Table 4.25.

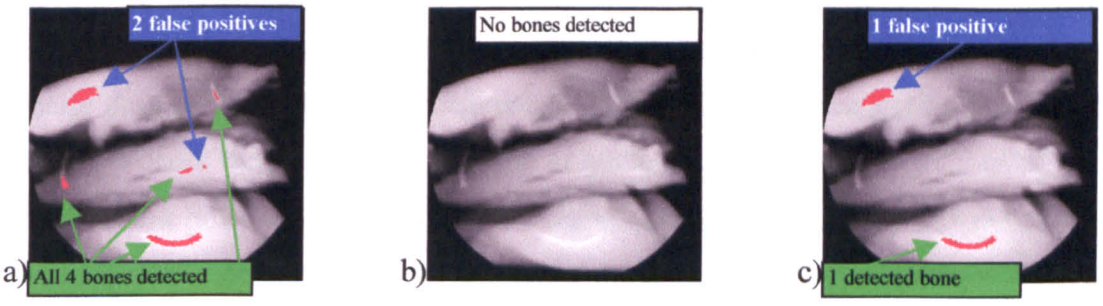


Fig.4.48 (ALL, LARGE) sensitivity result for image 4.47 b)
a) ALL sensitivity that generates 1 false positive; b) LARGE sensitivity (no bones are detected); c) (ALL, LARGE) sensitivity that detects 2 bones and improves the false positives percentage

Table 4.25 Fuzzy detection results

	ALL	SMALL	MEDIUM	LARGE
Accuracy	73.08%	90.71%	91.23%	89.30%
False positives	26.37%	4.39%	5.40%	5.20%
False negatives	0.55%	4.90%	3.37%	5.5%

A high sensitivity creates a large number of false positives, therefore creating more products to be rejected. One can play with a large number of sensitivity classes to suit the individual need, as it is explained in the next chapter.

4.5.2.3.6 Parameter sensitivity

Any FL approach is very sensitive with its parameters and the present architecture is no exception. However, parameter sensitivity is considered as being one of the advantages of an FL approach. It confers flexibility to the design and the architecture and this can be changed at any time during run-time, as opposed to a NN architecture that needs to be retrained again if any of its parameters are changed. It allows end-users to change fuzzy statements to fit new demands or new environment conditions. The parameter sensitivity in this case can be viewed as a form of flexibility.

In the present case, the parameters of the architecture consist of the defined points value of the MFs defined for the input fuzzy sets. Different MF were defined for the fuzzy sets and based on experiments and human direct observations optimal values were chosen (for the MF parameters) in order for the system to be robust and flexible.

4.5.2.3.7 Noise concerns

Since the architecture is based on FL it is quite tolerant to imprecise input data. Imprecise data can also be regarded as noise. As with any real-time application the food inspection process can be affected by noise at any stage of the process. In this late stage of classification noise is translated into corrupted feature values. How the performance of the algorithm performance changes in the presence of noise is presented in Table 4.26 and Figure 4.49. As it can be seen the change of performance is quite insignificant when the noise level is set at 3%, 5%, 10% and 15%.

Table 4.26 Noise levels and the change in FL approach performance

Noise level	3%	5%	10%	15%	20%	25%	30%
ALL	0.3%	1.2%	2.5%	6.7%	11.2%	18.2%	25.8%
SMALL	0.28%	0.9%	2.1%	6.4%	10.9%	17.8%	26.7%
MEDIUM	0.17%	0.8%	2.3%	6.4%	12.3%	17.7%	28.7%
LARGE	0.45%	0.34%	3.4%	7.1%	14.3%	18.3%	30.2%

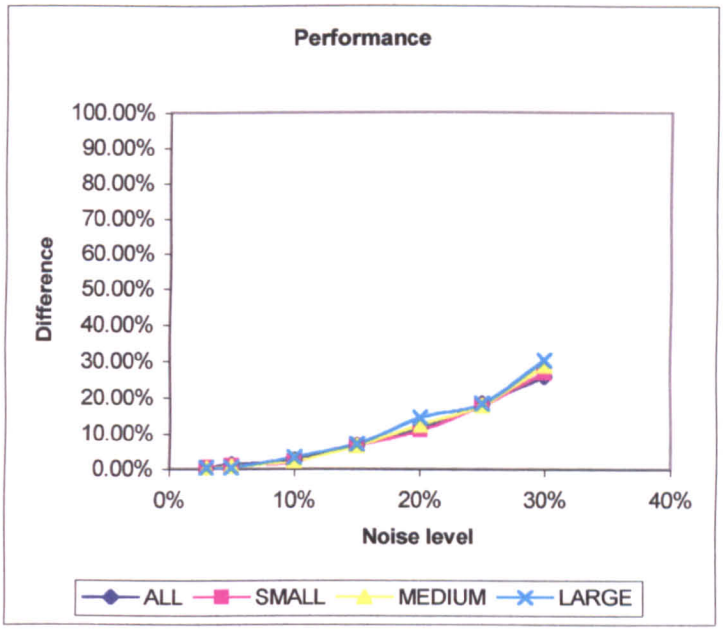


Fig.4.49 Change in the performance accuracy for different levels of noise

4.6 Summary and recommendations

Classical and artificial intelligence methods have been investigated in the context of image analysis. As shown previous in this chapter, the dual-band image taken from the meat product is first segmented into meaningful objects by using a HNN based segmentation algorithm. These objects need to be classified into foreign-bodies or normal product classes and a FL approach is proposed along with a classical and a NN-based classification approach. The HNN-based method of segmentation, along with further fuzzy logic high level detection algorithm performs well in the context of the specific inspection task. As demonstrated, real-time, noise sensitivity and flexibility are among the advantages of the developed X-ray imaging inspection system. However, in order to determine system reliability and high degree of performance an evaluation of the system is necessary.

"I didn't think; I experimented."

Wilhelm Roentgen

C H A P T E R 5

SYSTEM EVALUATION

5.1 Introduction

Once a classification system has been designed and implemented it needs to be evaluated. Classical techniques of evaluation are based on comparative studies between the proposed system and other existing systems. For many years, the performance of classification systems has been measured by the percentage of classification decision that proved to be correct. Most reported studies are for medical imaging applications and refer to diagnostic systems, where the system has to correctly classify patients into diseased or healthy. The performance is also known as diagnostic (in medical studies) or test (in other disciplines) accuracy. The major shortcoming of such an evaluation approach is fairly obvious: it depends strongly on test prevalence *[Metz, 1986], [Metz, 1989].* For example, if only 5% of the entire inspected population can contain foreign bodies, then a system can be 95% accurate simply by randomly calling 95% of the products as being normal.

Also, by measuring only the percentage of correct classifications the relative frequencies of false-positive (FP) and false-negative (FN) errors are not taken into account.

By introducing an evaluation procedure based on a pair of indices one can overcome the above-mentioned disadvantages. These indices are *sensitivity* and *specificity*. *Sensitivity* is the proportion of products with foreign bodies that test positive. *Specificity* is the proportion of products without foreign bodies that test negative. When speaking of sensitivity and specificity one can simply realise that they are synonymous to the true positives fraction (TPF) and the true negatives fraction (TNF) respectively. Some authors use the terms false positives fraction or rate (FPF) and false negatives fraction or rate (FNF). Since FPF and FNF represent conditional probabilities of frequencies with which actually positive and actually negative products are classified incorrectly, the following relations can be deduced:

$$FNF = 1 - TPF = 1 - sensitivity$$
$$FPF = 1 - TNF = 1 - specificity \tag{5.1}$$

Sensitivity and specificity describe how well the test discriminate between products with and without foreign bodies. However, the use of only a single measure, either sensitivity or specificity, is inadequate due to the lack of information about TNs and TPs, respectively. Therefore, the pair of the two measures has to be used in order to have meaningful evaluation results.

The sensitivity and specificity of a test depends on more than just the 'quality' of the test. They also depend on the definition of what constitutes an abnormal test. In Figure 5.1 one can see the idealized curves for both numbers of positive and negative result for the test (Table 5.1). Since the test does not distinguish normal product from abnormal products with 100% accuracy, the distributions overlap. The area of overlap indicates where the test cannot distinguish between the two classes – normal and abnormal products. In practice, one chooses a cutpoint (referred by some authors as a confidence threshold) above which the test is considered normal. The position of this cutpoint will determine the number of TPs, TNs, FPs and FNs. One can use different cutpoints for different situations in order to minimise one of the erroneous types of test result. The shapes of the probability distributions of the confidence in a positive or a negative test need not be known. In Figure 5.1, bell-shaped curves are used for

illustration purposes, but any shape for the distributions will be used in the same manner when a confidence threshold is chosen.

Table 5.1 Positive and negative tests

	Foreign body present	Foreign body absent
Test positive	True positives	False positives
Test negative	False negatives	True negatives

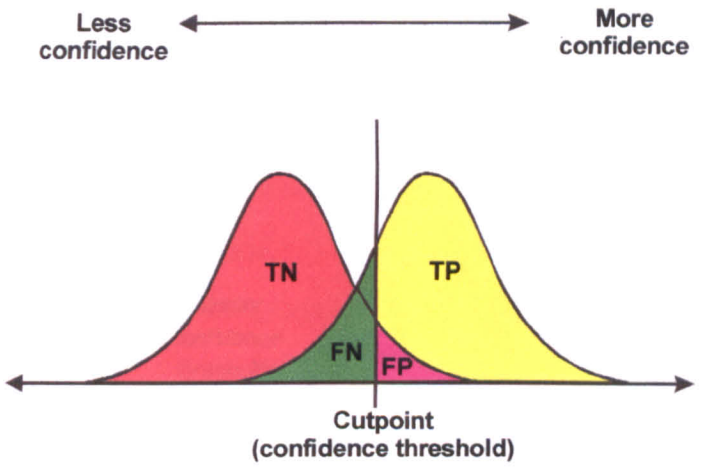


Fig.5.1 Confidence threshold separating "positive" decisions from "negative" decisions

The sensitivity and specificity of a classifier system will depend on the particular confidence threshold that the system decision unit is using, no mater if the decision is taken by a human operator or by an automated decision unit *[Metz, 1986]*. Sensitivity and specificity will change inversely as the confidence threshold is changed. If the confidence threshold is made less strict, then sensitivity will increase due to more products being classified as TPs. However, specificity will decrease because more FPs will appear (see equations 5.1). The pair TPF and FPF will increase or decrease together as the confidence interval threshold is changed.

5.2 ROC curves based evaluation

In order to compare the performance of two classification systems, one can use the sensitivity and specificity pair *[Woods and Bowyer, 1997]*. If the confidence interval

threshold, as defined in the previous section, is changed continuously, then a variety of (sensitivity, specificity) pairs will be generated. Those values can be plotted on a graph (see Figure 5.2). The smooth curve obtained is called a Receiver Operator Characteristic (ROC) curve *[Metz, 1986], [Metz, 1989]*. ROC analysis has been used in a large range of medical imaging studies. It is now an established method for the evaluation of medical imaging systems.

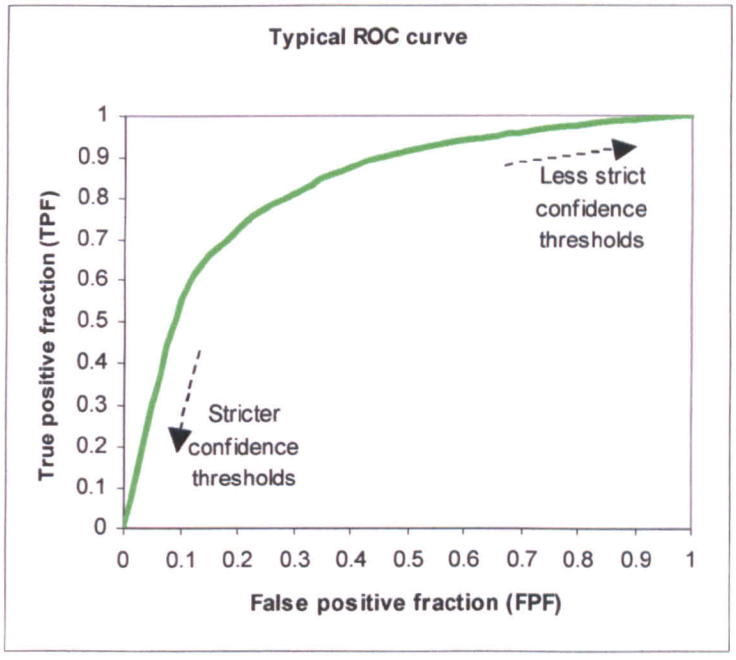


Fig.5.2 General ROC curve

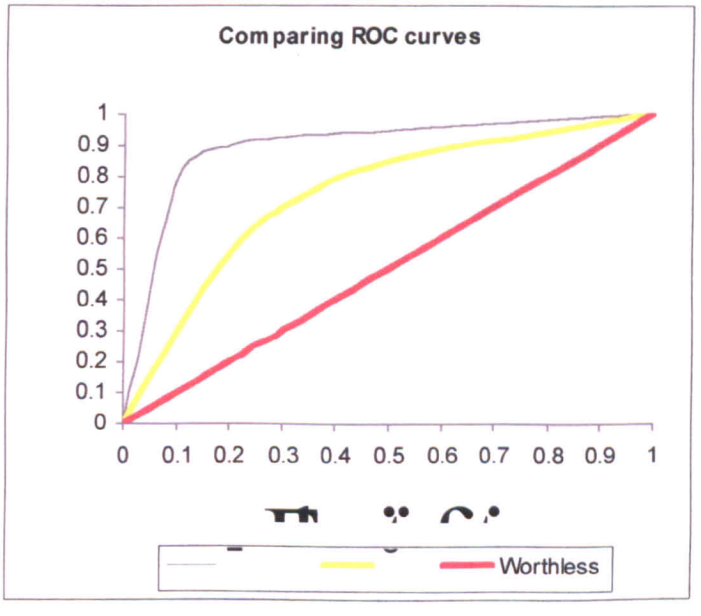


Fig.5.3 ROC curves comparison

An ROC curve demonstrates several things:

- Shows the trade-off between sensitivity and specificity; an increase in sensitivity will generate a decrease in specificity and the other way around;
- The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test; if the curve comes to the 45^0 diagonal than the performed test is less accurate;

Three ROC curves representing excellent, good and worthless tests plotted on the same graph are depicted in Figure 5.3.

In the following paragraphs ROC curves are used for the evaluation of the proposed automated inspection system, in comparison with a classical human operator based inspection system.

5.3 Human-based system evaluation

A human based inspection system has the property that the final decision is taken by a human operator. In most of the cases, the operator views the X-ray image of the inspected product on a monitor. Based on that image and his/her knowledge and experience the operator has the ability to reject or accept the inspected product.

However, humans are known to be subjective when making a classification. Also different levels of training and experience can render different results for different operators. Furthermore, human factors such as physical stimuli and the physical and mental state of the operator at the time of the inspection influence the outcome of the process. The surrounding environment can also make the operator-based inspection process a variable one (for instance, in different light conditions the decision can be changed dramatically due to change perceived on the monitor's screen). Experiments were performed in order to see how these human factors influence the inspection process.

5.3.1 Human factors influencing the system

Any system that involves human decision or the presence of human operators is open to subjective influences called human factors *[Reason, 1992], [Chapanis, 1996].* The amount of thinking and concentration a person exerts while using a system is called *mental workload.* The mental workload imposed on users by the environment in which

they use the system can exceed their abilities to use devices properly. Some authors refer to mental workload as mental "stress". Under high stress levels, the user is distracted and will have less time to make decisions.

Working environments can also limit the effectiveness of visual and auditory displays (lighted indicators, auditory alarms and other signals). For systems used in noisy environments, the user might not be able to notice alarms if they are not sufficiently loud or distinctive. Motion and vibration can also affect the ability of users to read displayed information *[Kaye and Crowley, 2000]*.

If the system will be used in low light conditions images might not be clear to the user. Other display information can be lost under brightly lit conditions due to insufficient contrast. With much surrounding distraction, important information could be missed.

Fatigue, stress, medication, or other temporary mental or physical conditions can temporarily affect ability levels of device users *[Kaye and Crowley, 2000]*. Typical human factors that can influence human performance and ability to use the systems are:

- General health and mental state
- Sensory capabilities (mainly vision for the inspection systems)
- Coordination
- Cognitive ability and memory (in order to concentrate on the inspection task or to gain useful experience)
- Motivation, and
- Ability to adapt to adverse circumstances

For example, older users might have difficulty remembering specific sequences for operation, using their hands to do tasks that require fine manipulation, or sensing system outputs such as auditory alarm sounds or information displayed visually.

In this study, experimental data was gathered from 4 persons: two "experts" that were trained on 150 X-ray images beforehand, one "medium" operator that was trained on 75 X-ray images and a "novice" that was only trained on 15 X-ray images just before the experiment. Each person was asked to operate the system for no more than 90 minutes, with a 5 minutes pause every 30 minutes. In essence, X-ray images were shown on the screen (different from the ones used in the training process) to the people and they were asked to either reject or accept the product (rejectance where the product contains a foreign body and acceptance where the product was "clean"). Most of the

people reported physical problem such as fatigueness, boredom, and eyestrain. These human factors influenced their performance in terms of correct classified products, but also from the FPs and FNs point of view. The results are shown in Tables 5.2-5.4 and Figures 5.4-5.6. After 90 minutes, even the "experts" dropped in accuracy down to 78-80%. It is expected, that at this rate, during a normal factory working cycle of 8 working hours, an operator's performance accuracy can drop down to 28%. Consequently, FNs and FPs values are increasing continuously.

Table 5.2 Human performance accuracy decay over time

ACCURACY	10 min.	20 min.	30 min.	40 min.	60 min.	90 min.
"Novice"	87%	85%	82%	77%	73%	65%
"Medium"	90%	89%	85%	81%	77%	72%
"Expert" 1	97%	96%	92%	90%	87%	78%
"Expert "2	96%	96%	94%	92%	87%	80%

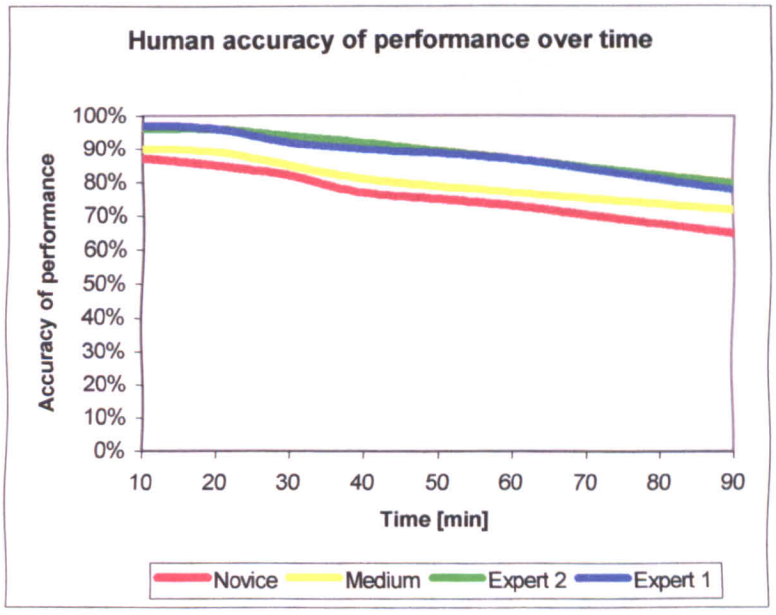


Fig.5.4 Human's performance accuracy decrease in time

Table 5.3 FPs over time for the human operators

FPs	10 min.	20 min.	30 min.	40 min.	60 min.	90 min.
"Expert" 1	1%	2%	4%	4%	5%	10%
"Novice"	5%	7%	9%	11%	13%	17%
"Medium"	4%	5%	7%	8%	10%	13%
"Expert" 2	2%	2%	3%	3%	5%	7%

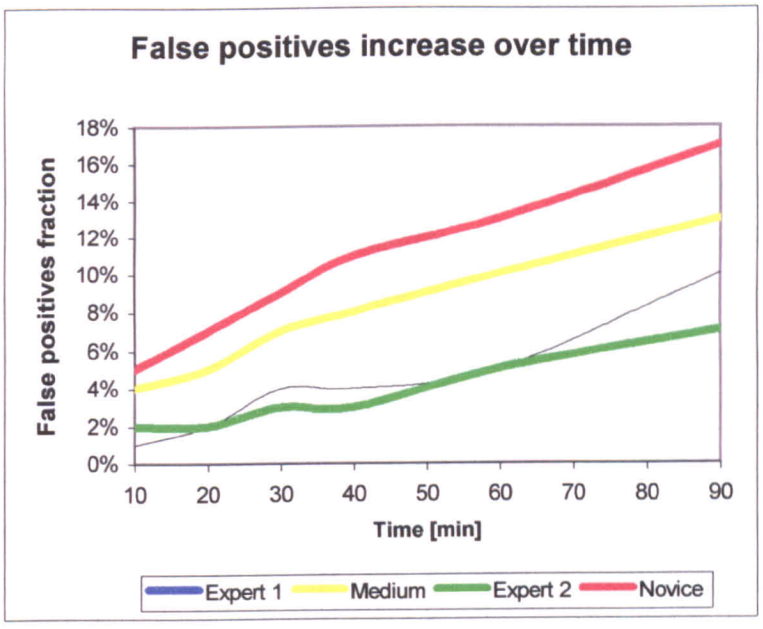


Fig.5.5 FPs increasing tendency with time

Table 5.4 FNs over time for the human operators

FNs	10 min.	20 min.	30 min.	40 min.	60 min.	90 min.
"Expert" 1	2%	2%	4%	6%	8%	12%
"Novice"	8%	8%	9%	12%	14%	18%
"Medium"	6%	6%	8%	9%	13%	15%
"Expert" 2	2%	2%	3%	5%	8%	13%

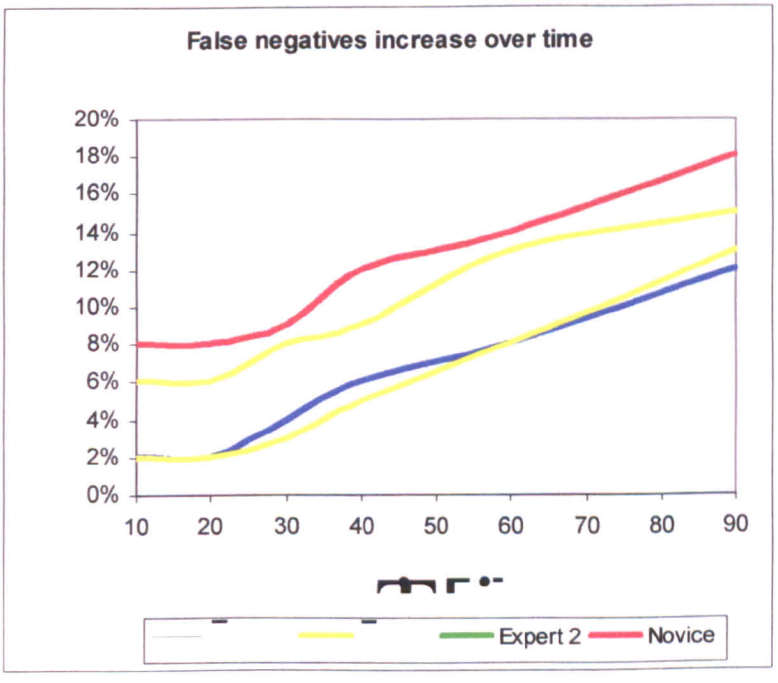


Fig.5.6 FNs increasing tendency with time

5.3.2 Results

In order to generate a ROC curve, human operators were asked to perform detection tasks after recovery time so that fatigue was minimised and may be rejected in optimal environmental conditions. Persons basically viewed an X-ray image of the inspected product on a monitor. If they considered the product free of foreign bodies then they accept the product; otherwise, the product is rejected. In the first case, a positive result of the inspection test is generated, whereas in the second case a negative result is obtained. However, in this way, only one point on a ROC curve will be obtained. Therefore, the human operators were asked to inspect the same sample of products (same sample of X-ray images were shown randomly on the monitor) with different levels of confidence: reject all products that incorporate at least one foreign body (less strict level of confidence), reject all products that have 2 foreign-bodies embedded within (high level of confidence), and reject all products that have more than 3 foreign-bodies embedded within (higher level of confidence). In total, operators were shown 150 X-ray images, from which 60 images of products that contained foreign bodies and 90 images of foreign body-free products. Sensitivity and specificity were measured for all these confidence intervals and using equations (5.1) FPF values were computed. The results are shown in Tables 5.5 and 5.6 for an "expert" operator and in Tables 5.7 and 5.8 for a "novice" operator.

Table 5.5 "Expert" operator results

Level of confidence	Sensitivity (TPF)	Specificity (TNF)
1 foreign body	0.98	0.87
2 foreign bodies	0.86	0.93
3 or more foreign bodies	0.66	0.99

Table 5.6 "Expert" operator computed FPF

Level of confidence	TPF	FPF
1 foreign body	0.98	0.13
2 foreign bodies	0.86	0.07
3 or more foreign bodies	0.66	0.01

Table 5.7 "Novice" operator results

Level of confidence	Sensitivity (TPF)	Specificity (TNF)
1 foreign body	0.96	0.83
2 foreign bodies	0.82	0.90
3 or more foreign bodies	0.60	0.95

Table 5.8 "Novice" operator computed FPF

Level of confidence	TPF	FPF
1 foreign body	0.96	0.17
2 foreign bodies	0.82	0.10
3 or more foreign bodies	0.60	0.05

Along with the obvious points generated by (TPF=0;FPF=0) and (TPF=1;FPF=1), the pairs of TPF and FPF values from Tables 5.6 and 5.8 can be plotted in order to generate the ROC curves for the "expert" and the "novice" human-based inspection system performance evaluation (Figure 5.7).

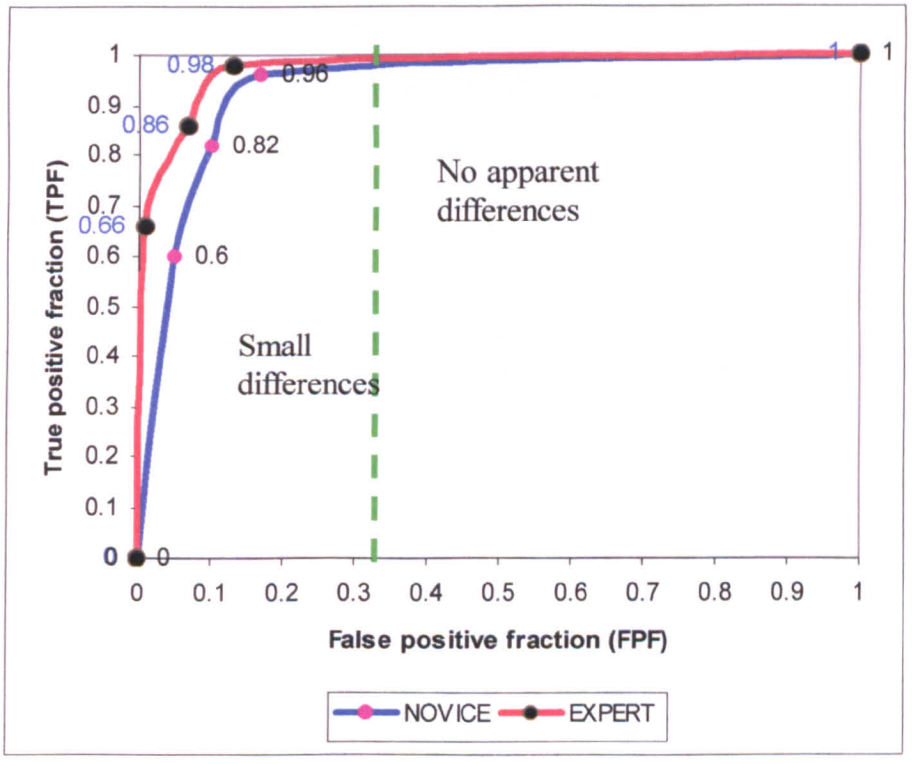


Fig.5.7 ROC curves for "expert" and "novice" human operators

As can be expected, an "expert" operator shows a slightly greater discrimination capacity as the ROC curve suggests. The difference between "expert" and "novice" performances suggested by the ROC curves is greater when, at the same confidence

threshold, FPF is low and the TPF is high (towards the left hand of the graph). When the FPF is greater than 0.3, then the difference between ROCs shows no apparent difference between operators.

Using the values from Tables 5.2 and 5.4, one can compute the FNF and FPF values at different times to see how the decay in time of human performance affects the ROC evaluation. Two cases were considered and analysed here: 30 minutes (first part of the experiment) and 90 minutes (end of the experiment). The results are shown in Tables 5.9 -5.12 for the "Expert" and the "Novice" human operators.

Table 5.9 "Novice" operator computed TPF and FPF after 30 minutes

Level of confidence	TPF	FPF
1 foreign body	0.25	0.78
2 foreign bodies	0.18	0.67
3 or more foreign bodies	0.14	0.49

Table 5.10 "Novice" operator computed TPF and FPF after 90 minutes

Level of confidence	TPF	FPF
1 foreign body	0.32	0.62
2 foreign bodies	0.27	0.53
3 or more foreign bodies	0.22	0.39

Table 5.11 "Expert" operator computed TPF and FPF after 30 minutes

Level of confidence	TPF	FPF
1 foreign body	0.17	0.9
2 foreign bodies	0.11	0.79
3 or more foreign bodies	0.05	0.6

Table 5.12 "Expert" operator computed TPF and FPF after 90 minutes

Level of confidence	TPF	FPF
1 foreign body	0.24	0.76
2 foreign bodies	0.19	0.67
3 or more foreign bodies	0.13	0.51

As depicted in Figures 5.8 and 5.9 , the ROC curves at later times tend to move towards the 45^0 diagonal, thus the human performance is less and less accurate.

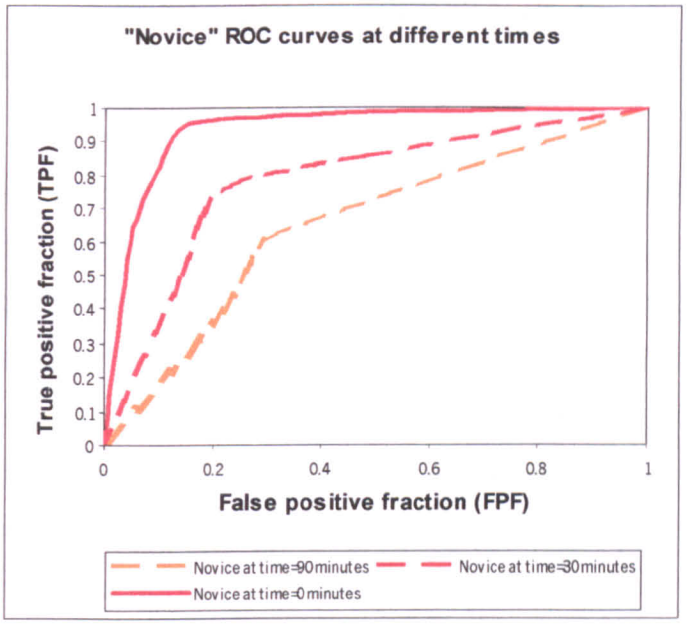


Fig.5.8 ROC curves variation with time for "novice" human operators

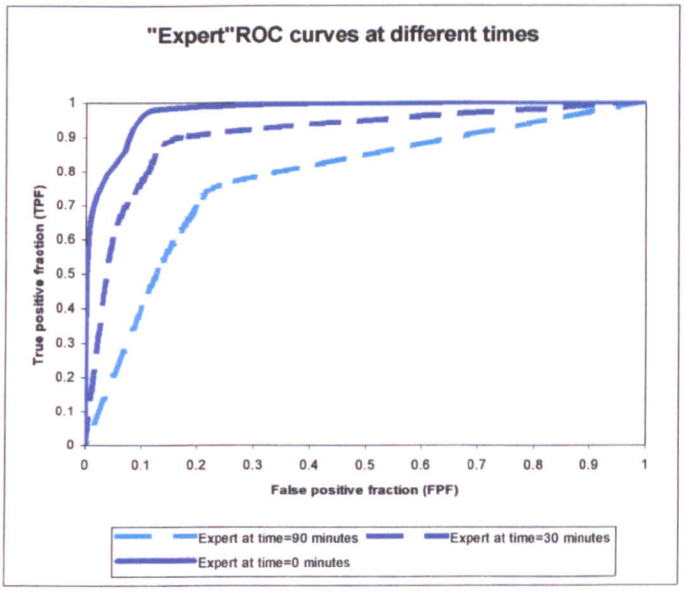


Fig.5.9 ROC curves variation with time for "expert" human operators

5.4 Proposed automated system evaluation

Generally, a classification system has a single output. However, by manipulating one or more of the architecture parameters the proposed system can generate many points on a ROC curve. The commonly accepted method of generating ROC points is to vary a threshold over the range of the output (0.0 to 1.0) *[Swets et al., 1979], [Woods and Bowyer, 1997]*. The basic equation is:

$$output = \begin{cases} true, \ if \ greater \ than \ or \ equal \ to \ the \ chosen \ threshold \\ false, otherwise \end{cases} \quad (5.2)$$

In the present case, the first fuzzy module's output (refer to paragraph 4.5.2.3.2) can be thresholded to correspond to successive higher levels of sensitivity. In essence, when thresholding the system's output, one changes the probability of a correct decision. If the threshold is set at 0.95, then the system will take the decision with a higher degree of confidence (95% probability of being the true answer). In contrast, when the threshold is taken as 0.2, then a very low confidence is used (20% probability of being the true answer). A sample of 150 (60 positive cases and 90 negative cases) images was used to generate the ROC curve for the performance of the automated system. The system was presented with an image and its answer is taken as a positive or a negative answer according to the confidence threshold by using equation (5.2). When the confidence threshold is set at the lowest value, the system emulates a human operator with a high confidence. This inspection task is similar with the previous tasks performed by the test operators who were asked to detect all the foreign bodies present in the product (X-ray image). In contrast, a system with a low output threshold mimics a human operator doing an inspection task with a low confidence (detect only large foreign bodies).

The threshold values and the results of decisions of the system decision are shown in Table 5.13.

Table 5.13 Threshold values and system results

FUZZY OUTPUT	0.5	0.6	0.7	0.8	0.9	TOTAL
Positive cases (products with foreign bodies)	28	18	7	4	3	60
Negative cases (normal products)	1	9	15	23	32	90

Table 5.14 Results for a confidence threshold of 0.5

Threshold	≤0.5	>0.5	TOTAL
Positive cases (products with foreign bodies)	28	32	60
Negative cases (normal products)	1	89	90

Table 5.15 Results for a confidence threshold of 0.6

Threshold	≤0.6	>0.6	TOTAL
Positive cases (products with foreign bodies)	46	14	60
Negative cases (normal products)	10	80	90

Table 5.16 Results for a confidence threshold of 0.7

Threshold	≤0.7	>0.7	TOTAL
Positive cases (products with foreign bodies)	53	7	60
Negative cases (normal products)	25	65	90

Table 5.17 Results for a confidence threshold of 0.8

Threshold	≤0.8	>0.8	TOTAL
Positive cases (products with foreign bodies)	57	3	60
Negative cases (normal products)	48	42	90

If one uses a cutoff of 0.5 then the sensitivity is 0.46 and the specificity is 0.98; if the threshold is changed to 0.6 then the sensitivity is 0.76 and the specificity is 0.89. For a threshold of 0.7 then the sensitivity is 0.88 and the specificity is 0.72 and for a threshold of 0.88 then the sensitivity increases to 0.95 while the specificity decreases to 0.47 (results are summarised in Tables 5.14-5.18).

By using equations (5.1), one can derive the FPF values from the TPF (Table 5.19).

Table 5.18 Sensitivity and specificity for the computer based inspection system

Confidence threshold	Sensitivity (TPF)	Specificity (TNF)
0.5	0.46	0.98
0.6	0.76	0.89
0.7	0.88	0.72
0.8	0.95	0.47

Table 5.19 Computer TPF and FPF

Confidence threshold	TPF	FPF
0.5	0.46	0.02
0.6	0.76	0.11
0.7	0.88	0.28
0.8	0.95	0.53

Using the values in Table 5.19, a ROC curve of the system can be plotted (Figure 5.10).

The plotted ROC curve suggests quite a good discrimination capacity of the proposed system since the ROC curves is not approaching the diagonal curve. However, in order to make a comparison between human performance and system performance, one can analyse the difference between their corresponding ROCs.

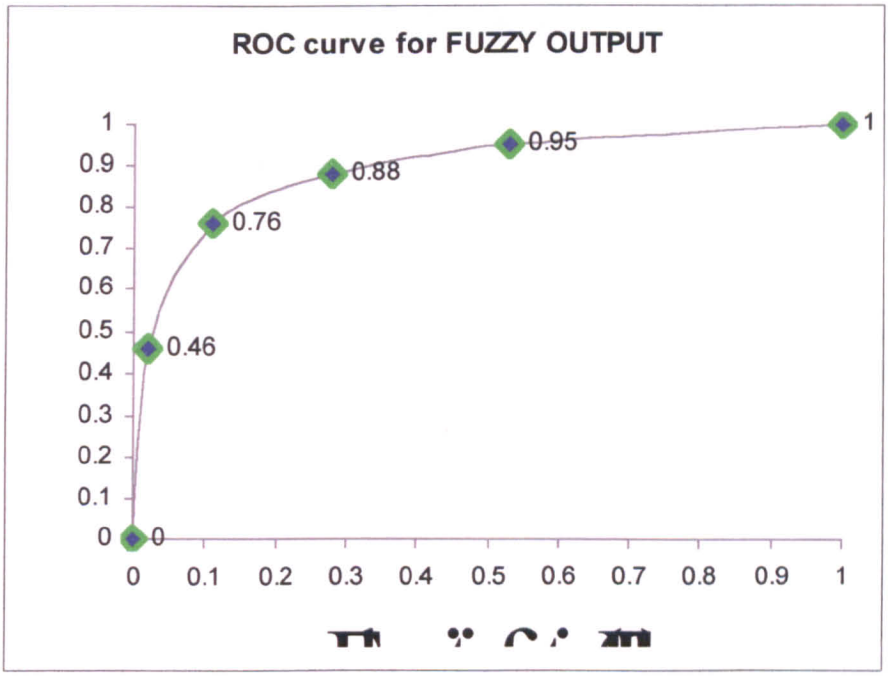


Fig.5.10 Computer ROC curve

5.5 Comparative study

Figure 5.11 shows the resulting ROCs for an "expert" human operator at time=0, 30 and 90 minutes, for a "novice" human operator at time=0, 30 and 90 minutes and for an automated computer driven inspection system for the same sample of X-ray images of chicken breast meat. With all the uncertainties present in the real world, when emulating human knowledge on a computer, one expects that system performance to be below human performance. The graph is showing just that, for most of the points on the ROCs. However, as explained in paragraph (5.3.1) any human operator based system performance is influenced by numerous human and environmental factors. Therefore, it is safe to assume that the ROC curves for the human operators tend to move towards the diagonal from the lower left hand corner to the upper right hand corner (as depicted in

Figure 5.11). A computer-based system, however, is uninfluenced by subjective factors and therefore its performance will remain unchanged over time.

The context of the application is to find all possible defects. This evaluation is biased towards humans in that the sample used has a high rate of TPs. In practice this rate is much less and leads to much less effective human performance (fatigue effects increase due to boredom). One would expect infrequent failures to occur and the system would outperform human operators. The exploration of this in detail is outside the scope of this thesis, although the ROC method can be applied.

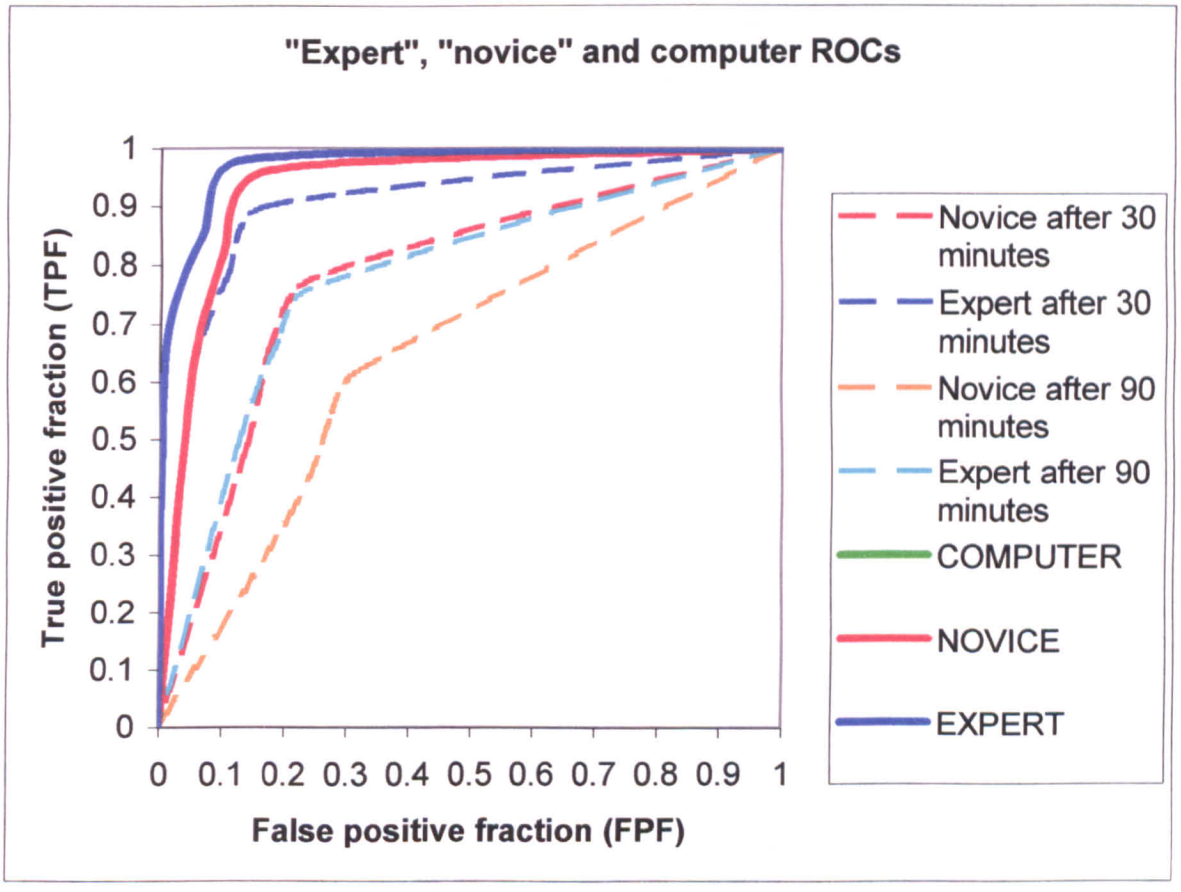


Fig.5.11 Comparison between human and computer ROCs

5.6 Summary and recommendations

The successful application of image processing method based on AI techniques in a specific context was demonstrated. The next goal is to propose a generic X-ray imaging system based on this method.

"For every problem, there is one solution which is simple, neat and wrong."

Henry Louis Mencken
1880-1956

C H A P T E R 6

A GENERIC X-RAY IMAGING INSPECTION SYSTEM

6.1 Introduction

A generic system is depicted as in Figure 6.1. It consists of a black box, completely transparent to the end-user that realises a mapping between its input space and its output space. Hence, the black box is a function that deals with all computations necessary in order to generate the desired output from the input space.

Fig. 6.1 General description of a generic system

The proposed system for the detection of foreign bodies in a meat product in this work (chapter 4 and 5) has high flexibility. The user can optimise its parameters in order to meet their specific inspection needs. Furthermore, such a system allows an end

user to choose a particular confidence threshold when the final inspection decision is taken. Therefore, such a system can be regarded as the basis for a generic X-ray imaging inspection system.

6.2 A proposed generic X-ray imaging inspection system

6.2.1 System's architecture

The proposed system (Figure 6.2) is a summary of the sub-systems proposed in chapter 4. It comprises the following units or systems:

- *Image acquisition system* – means of generation of X-ray images of the inspected product and means of transmitting them to a computer;
- *Image pre-processing system* – techniques for enhancing the X-ray images for intermediate and high image processing (contrast enhancement, background removal, noise removal, etc.);
- *Image segmentation system* – a Hopfield Neural Network module that partitions the X-ray image into meaningful classes for further higher level inspection;
- *Object extraction unit* – extracts objects from the previously segmented image;
- *Feature extraction system*: comprises two modules that extract geometrical features and grey-level based statistical features respectively, of each extracted object;
- *High level detection system* – a fuzzy logic based system for the higher inspection level; it consists of two fuzzy modules that use fuzzy inference systems based on rules about the above mentioned extracted features.

A summary of the inspection process is presented below:

General X-ray imaging inspection process

Step 1

Acquire the X-ray image or images of the inspected product

Step 2

Low-level image processing of the resulting image or images

a) X-ray image or images **pre-processing**

b) X-ray image **segmentation** using a Hopfield Neural Network (HNN)

c) **Object extraction** for the obtained segmented image

Step 3

High-level detection of possible faults or defects of the product

(a defect is considered here as a generic notion; it represents any possible problems with the inspected product that can make it not useful for its designed purpose)

a) **Feature extraction** for the extracted objects

b) **High-level fuzzy logic based detection** of possible faults or defects

Step 4

Final product **acceptance** or **rejectance**

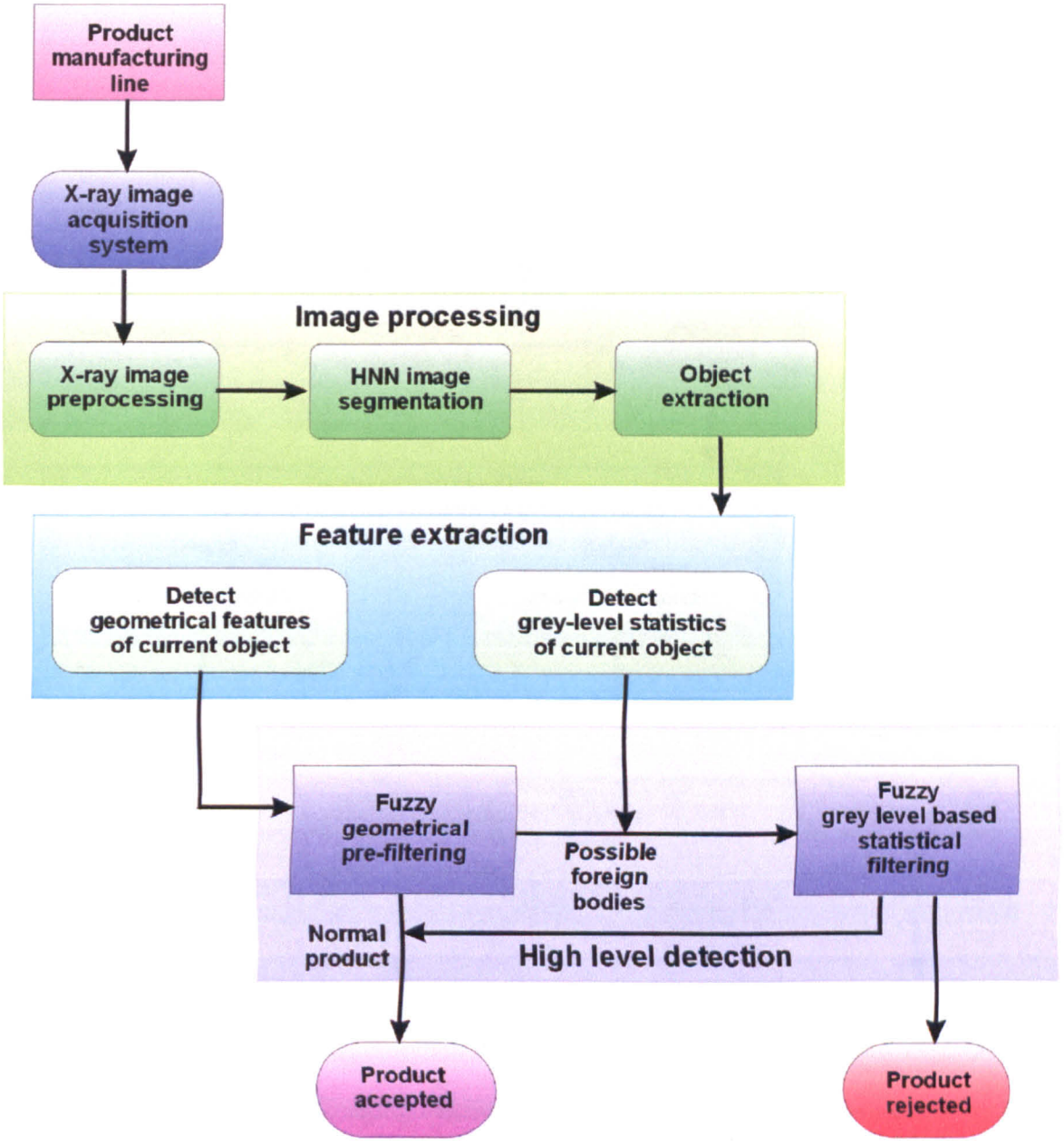


Fig. 6.2 Generic X-ray imaging inspection system

Human experience has to be incorporated into the design of such a system. The knowledge obtained using experiments by human expert is gathered into a common database that is used along the entire inspection system. The knowledge database contains data about the inspected product (such as physical and chemical characteristics), data about the possible faults and defects (such as the presence of foreign bodies and their characteristics, cracks, porosities, etc.) and any other information direct or indirect related to the inspection process.

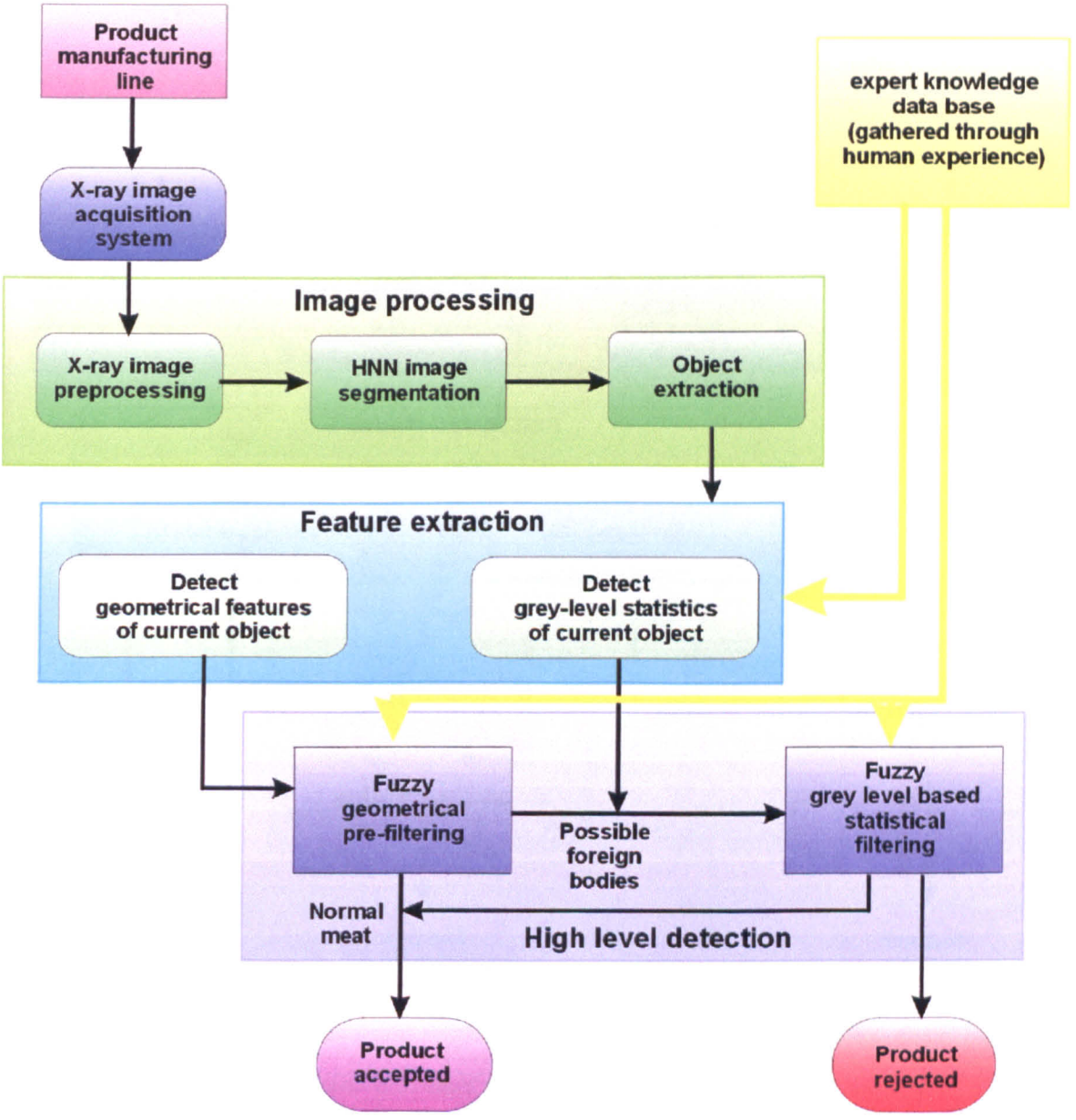


Fig. 6.3 Generic X-ray imaging inspection system incorporating human experience (yellow module and lines)

In Figure 6.3 one can see that the knowledge database is mainly used in the feature extraction process and in the high-level detection of possible faults or defect. Based on the product knowledge gathered through human experience, the system must be able to extract the appropriate features from the segmented objects in order to facilitate the high-level detection process. This task makes use of the same database, as well.

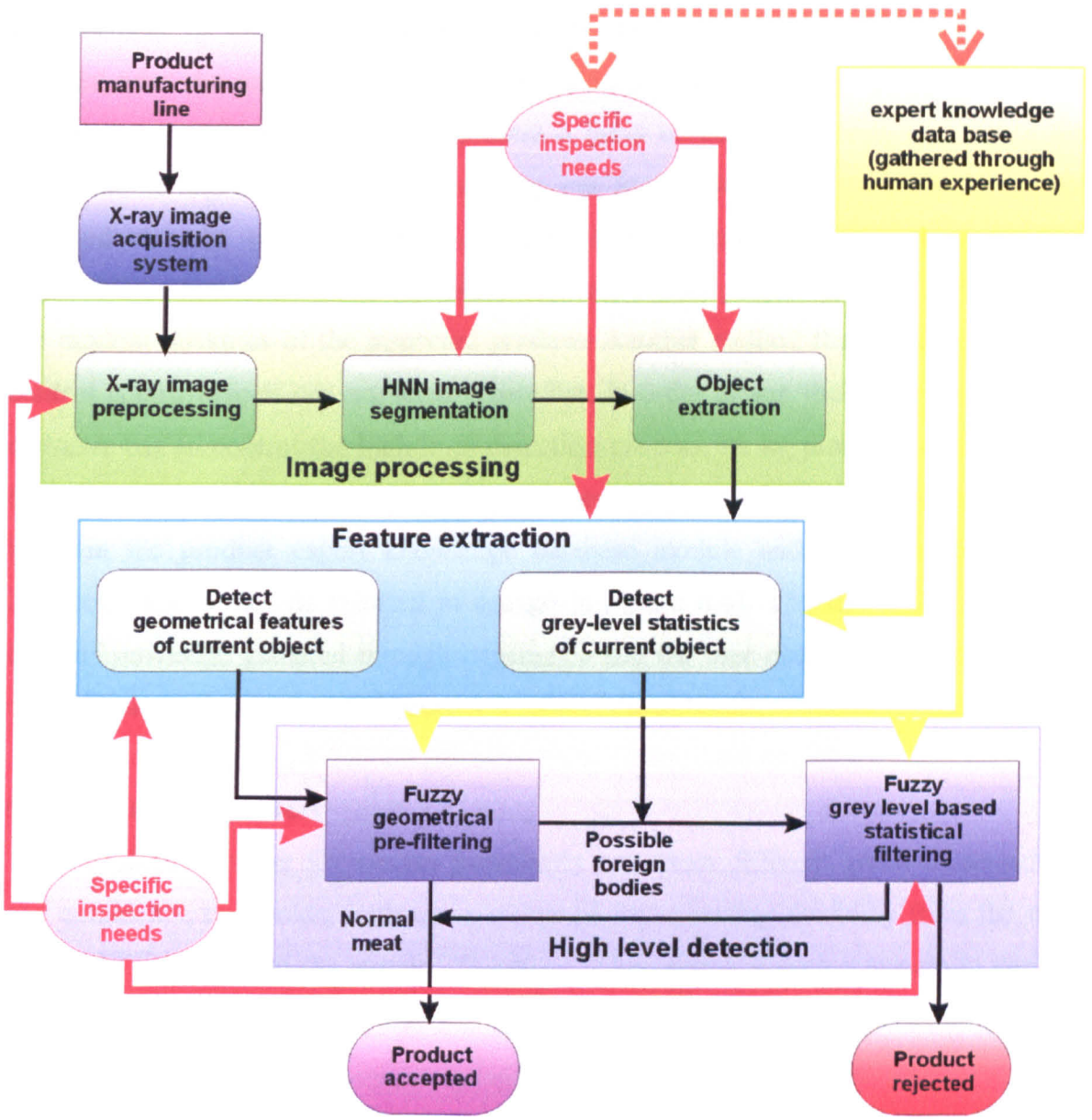


Fig. 6.4 Generic X-ray imaging inspection system incorporating human experience (yellow module and lines) and end-user specific inspection needs (red module and lines)

Being a generic system, one has to be able to implement the system for its own gain and use within a specific inspection process. The end-user has to be able to

incorporate preferences in the inspection process. As a result, a module for specific inspection needs is used to obtain flexibility from an end-user point of view, as depicted in Figure 6.4. The user preferences have to be taken into account in most of the inspection stages, from the low-level image pre-processing to the high level detection of faults. In the image processing part of the system, user preferences come heavily into play, starting with the pre-processing unit where one user wants a specific way of enhancement of the X-ray images to suit its needs. Image segmentation takes into account what the user wants to detect. For instance, when detecting the die-attach voids on electronic chips, products that possess a fairly constant thickness, one wants the images to be segmented into only 2 classes, one for the defects and one for the normal attached die. In contrast, when detecting foreign bodies from a meat product segmentation into two classes will miss a large number of possible foreign bodies due to the random thickness of the inspected product. Another method that the user wants to control is how objects are extracted. Therefore, in order to suit their specific needs the end-user has to control the high-level detection process. At all times, to achieve more flexibility and tuneability of the system, there is a two-way communication link between the product expert knowledge database module and the end-user specific inspection needs module (plotted in orange in Figure 6.4). The link makes sure that human knowledge gathered through experience and the user needs are mixed and used either sequentially or in parallel in order to obtain the optimum system's performance.

6.2.2 System parameters

Different values for system parameters can mean different levels of sensitivity and confidence thresholds; both are a means of incorporating flexibility from the end-user point of view into the system. Therefore, when applying such a system in various industrial or medical fields, one has to choose parameters and values in order to meet performance requirements.

System parameters can be summarised as follows:

- Parameters for the Hopfield Neural Network, and implicitly for the segmentation process. The most important parameter here is the number of classes into which one wants to segment the acquired X-ray images. Moreover, due to its meaning and context dependence, the definition of

the network energy is regarded as a parameter. How that energy is defined can render completely different results in different applications;

- Input measures for the fuzzy processing modules can also be considered as parameters, since they strongly depend on the specific field of application for the inspection system;
- The fuzzy sets for the fore-mentioned input measures must also be considered as parameters since they lead to different sensitivity levels for the inspection system. They are also dependant on the application field of the inspection system.

All or a selection of the above parameters can be modified in order to meet specific detection criteria. This constitutes the power of the proposed inspection system and confers the characteristic of generality on it. The proposed inspection system can therefore be viewed as a general framework that can be applied to different inspection task in various fields by simply modifying some or all of its parameters. Thus, the inspection system can be viewed as a black box that receives input parameters highly dependant on human experience and knowledge but also on human specific needs for the inspection task at hand and generates the desired output (see Figure 6.5). In general, the desired output consists of two values or classes: normal product and defective product.

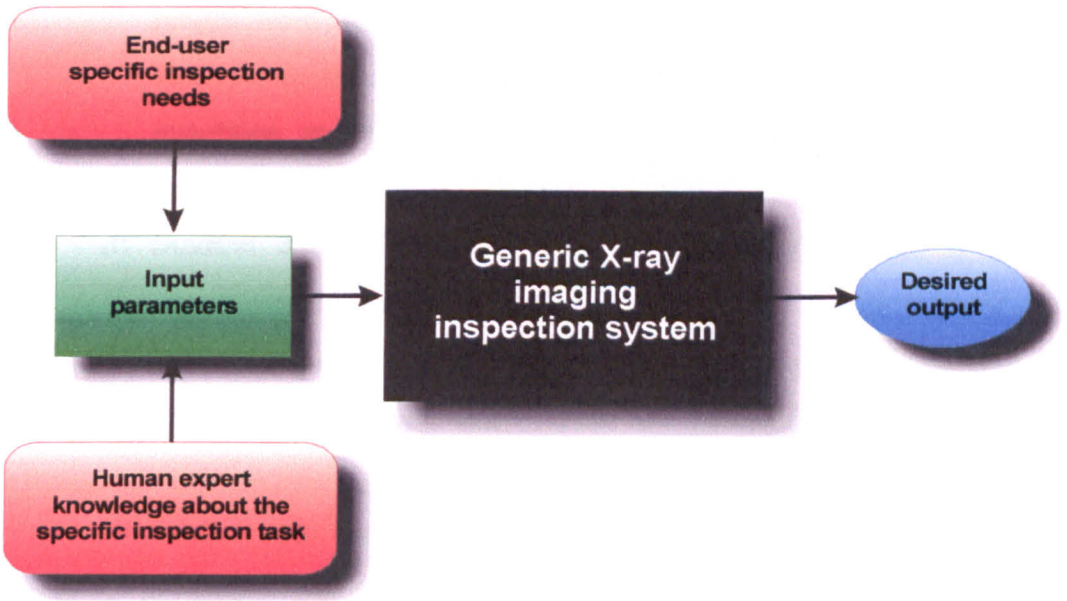


Fig.6.5 Generic inspection system as a black box

Example of how these parameters can be changed in different fields of application of an inspection system is demonstrated in the following paragraphs. Conventions from Chapter 4 are used in the following paragraphs. Some fuzzy input measures are used exactly as defined in paragraph (4.5.2).

6.3 Industrial X-ray imaging inspection applications

A great variety of X-ray equipment currently available offers a comprehensive range of penetrating power that can satisfy virtually any industrial requirement. These applications can range from irradiation studies, electronic circuitry and alloy casting inspection right through to the inspection of welded heavy steel structures.

Among the applications of X-ray imaging inspection techniques are:

a) Aviation and aerospace – X-ray inspection is one of the main quality assurance techniques used in the fields of aviation and aerospace, both in production and general repair and maintenance. Proper inspection of welds, of castings and materials like glass or carbon fibre reinforced plastics are an essential need for maximum safety.

b) Electrical engineering industry – X-ray inspection is applied to ensure the quality of the production of batteries, printed circuit boards (PCB), ball grid arrays (BGA), intergrated circuits, etc.

c) Automotive industry – The application of X-ray inspection techniques ranges from the inspection of light alloy castings to the inspection of steel products. X-ray imaging provides economical inspection of large quantities and this is well suited for the automotive manufacturing industry.

d) Medical devices – The inspection of medical devices is of critical importance to the their successful use. Safety is the primary concern here, where even the smallest defect can lead to a fatal situation for the subject. An artificial heart valve for instance, can be inspected for potential cracks or interruption in its contained circuitry.

e) Nuclear engineering – In the nuclear field, X-ray inspection has become a widely used technique in ensuring the safety of nuclear components. Reactor chambers and fuel vessels, nuclear plant thick concrete walls, are of vital importance.

Four applications are considered below. The object of the thesis is the applicability of an inspection system for the food industry and furthermore how the generic features emanating from the work may be applied in more general nature. Thus, a small number of X-ray images were obtained, (images available in Internet databases) to illustrate how the proposed generic inspection system can be applied in various application fields.

6.3.1 Application 1: Intelligent Composite Materials

Composite materials consist of two or more materials combined in such a way that the individual materials are easily distinguishable. A common example of a composite is concrete. It consists of a binder (cement) and a reinforcement (gravel). Adding another steel reinforcement (rebar) transforms concrete into a three-phase composite. The individual materials that make up composites are called constituents. Most composites have two constituent materials: a binder or matrix, and a reinforcement. The reinforcement is usually much stronger and stiffer than the matrix, and gives the composite its good properties. The matrix holds the reinforcements in an orderly pattern. Because the reinforcements are usually discontinuous, the matrix also helps to transfer load among the reinforcements. In the present study, a composite material was obtained using a polymer resin and a hardener as the reinforcement.

The problem of inspection of the composite materials is one of detection of non-uniformities in the X-ray images taken from the material. These non-uniformities can appear due to uneven mixing of the constituents polymers and hardeners, and therefore, one can find areas that contain only hardeners, or only polymers or uneven distribution of those across the full body of material.

Since the resulting composite material has to have a uniform viscosity the resulting X-ray image has to be uniform from the grey-level point of view. Possible air bubbles or non-uniformity of the composite material will appear on the image as areas with either a higher contrast or a lower contrast than the surrounding background. Therefore, HNN segmentation has to be performed into two classes: one class for the defects and the other one for the normal material. Once the segmentation process is finished, objects are extracted from the segmented image. These objects are then analysed and a final decision is taken whether they are or not defects by the high-level

fuzzy logic detection system. Examples of X-ray images taken from different samples of the same composite material are shown in Figure 6.6.

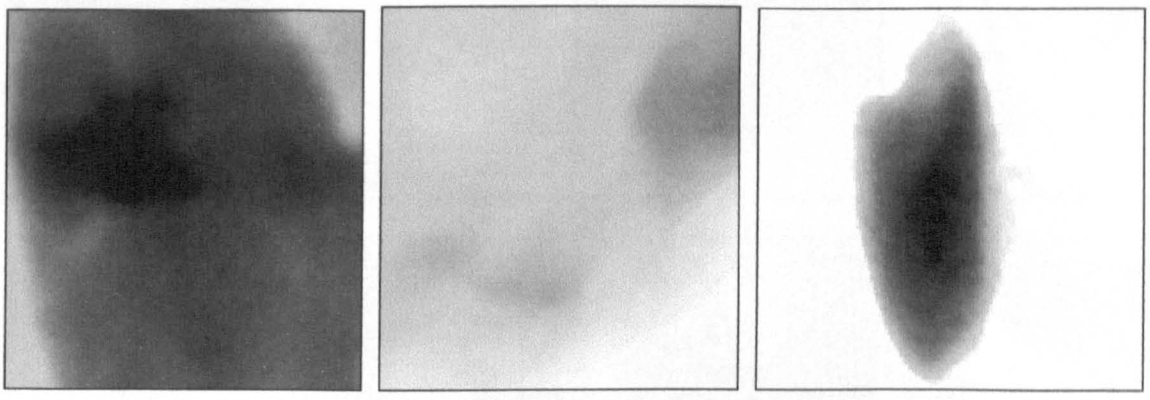
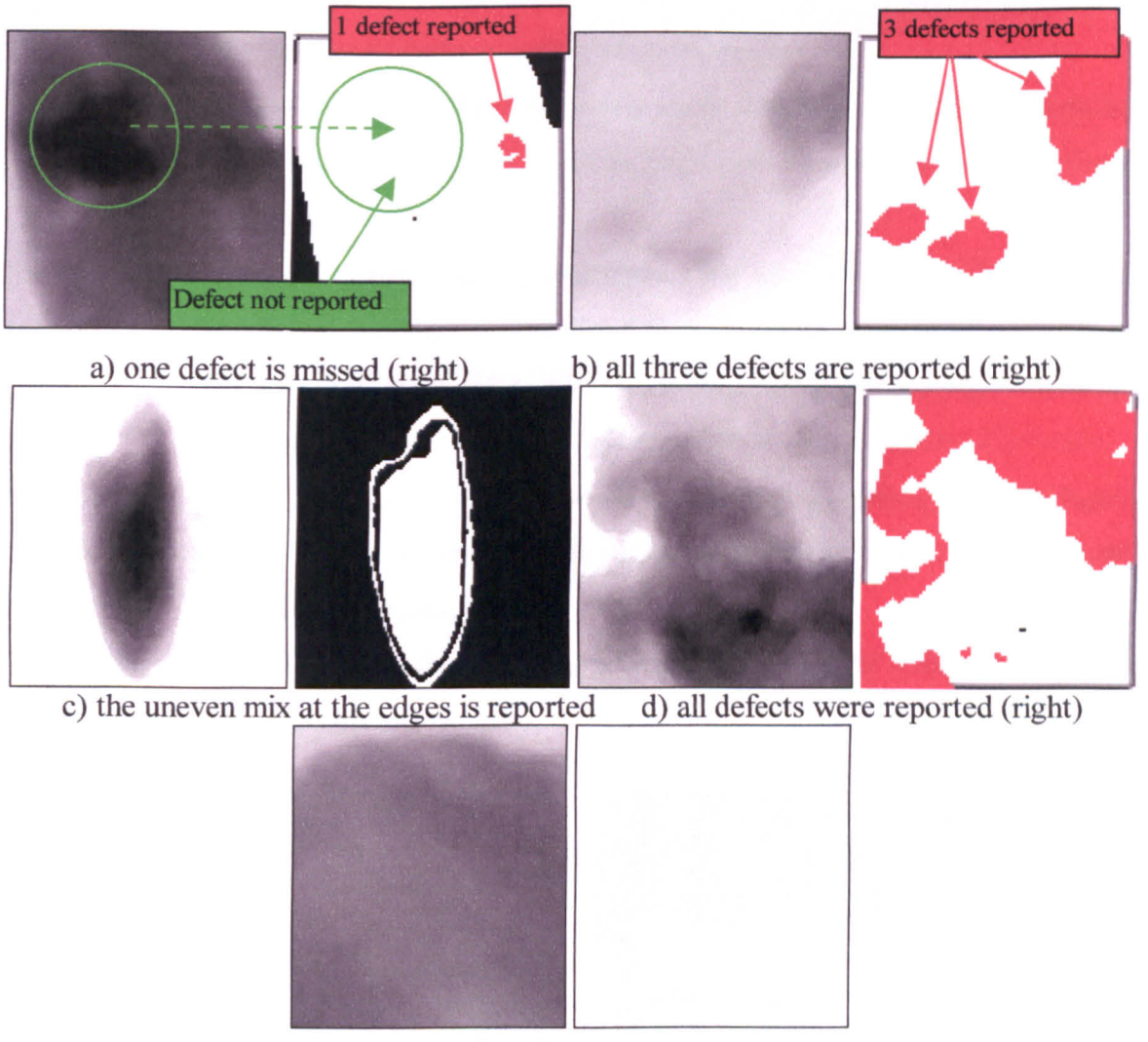


Fig.6.6 X-ray images of composite materials

The system was implemented for five X-ray images: 4 images with defects and 1 image without defects (Figure 6.7). In all cases one has to take into account the fact that the composite material always had a lower thickness near the edges and that should not be considered as faults.

The basic principle for the fuzzy rules was: if the average grey-level difference between current object and its surrounding background is high and it is not on the edge of the image then the object is a defect; otherwise, the object is normal material. In this case, there is no need for a pre-filtering module, since no geometrical features are extracted from the object. If one wants to measure the physical properties of the defects then one should use the geometrical fuzzy-module. The second fuzzy grey-level statistics module had two input measures: LOCATION (with two fuzzy sets: EDGE and MIDDLE) on the image of the current object and the DIFFERENCE (with two fuzzy sets: LOW and HIGH), measure defined as in (4.5.2.3.4.1). The output of the system had two fuzzy sets: DEFECT for when an object is classified as a defect and MATERIAL when the object is classified as normal material. There were four fuzzy rules for the implemented Mamdani-type fuzzy inference system:

1. if DIFFERENCE is HIGH and LOCATION is MIDDLE then OUTPUT is DEFECT
2. if DIFFERENCE is LOW and LOCATION is MIDDLE then OUTPUT is MATERIAL
3. if DIFFERENCE is HIGH and LOCATION is EDGE then OUTPUT is MATERIAL
4. if DIFFERENCE is LOW and LOCATION is EDGE then OUTPUT is MATERIAL



a) one defect is missed (right) b) all three defects are reported (right)

c) the uneven mix at the edges is reported d) all defects were reported (right)

e) no defect is reported (right)

Fig.6.7 X-ray images of composite materials and their corresponding system-output; objects in red are reported as defects

a),b),c),d) composite materials with defects; e) composite materials without defects

As shown in Figure 6.7 the system performs well. However, the level of detection is quite low in some cases (Figure 6.7 a where a defect is missed) and therefore more experiments need to be performed in order to render appropriate system parameters.

6.3.2 Application 2: Inspection of welds and castings

6.3.2.1 Inspection of welds

Weld metal defects are normally caused by a combination of high stresses across the weld and metallurgical weakness of the filler metal. These defects can either appear during the welding process or after.

There are a number of types of weld defect categories:

- *gas pores*, defects that occur as a result of gas entrapment in the molten metal during a weld or through improper cleaning of the joint during preparation of the weld;
- *incomplete fusion and penetration*, that can arise from poor welding conditions. Fusion refers to the degree to which base metal surfaces are fused to their metal filler, while penetration refers to the degree to which metal surfaces have been melted together to form the throat of a weld;
- *cracks* may occur in either the weld or base metals being joined. There is a wide variety of crack types (transverse, longitudinal and so on), each with sub-definitions.

As far as defect detection goes, it is possible to reliably recognise all these defects using X-ray inspection techniques.

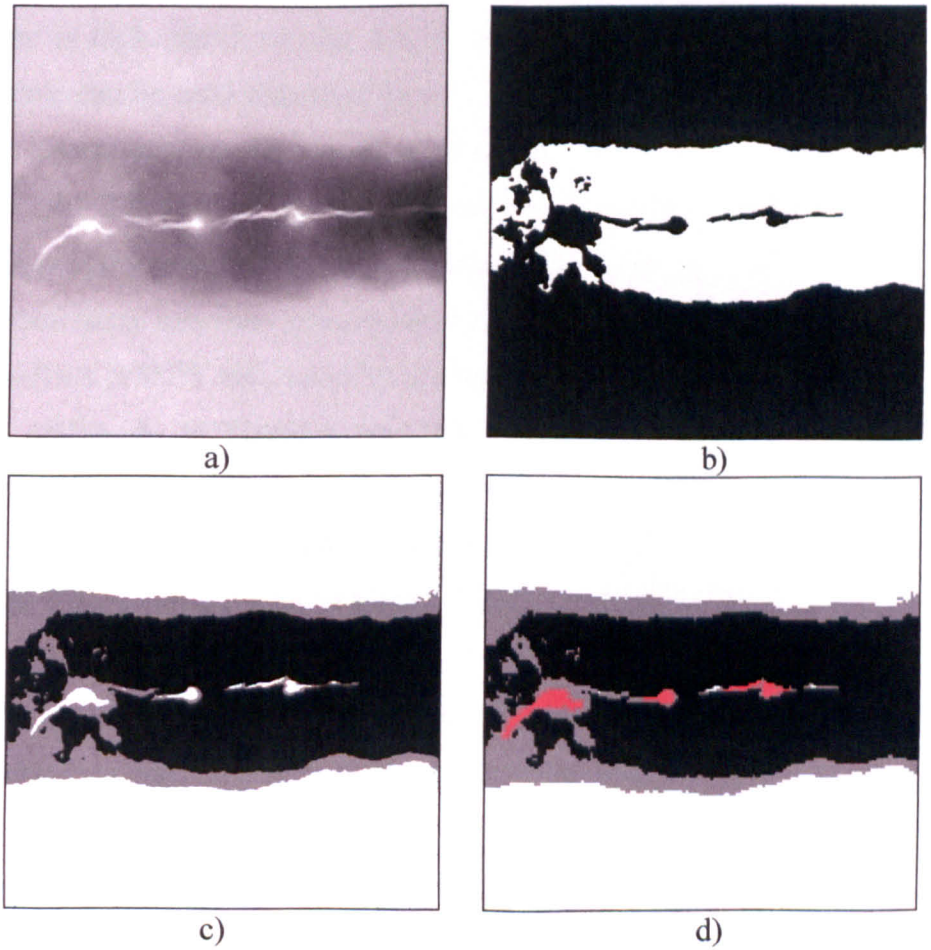


Fig.6.8 X-ray image of one weld with cracks

a) acquired X-ray image; b) segmentation into 2 classes; c) segmentation into three classes; d) results of the detection of cracks process (cracks depicted in red)

For the detection of cracks in welds, one wants the images to be segmented into two classes: one class for defects or cracks and another class for normal material. In Figure 6.8 a weld with cracks is depicted. As can be seen, the cracks have quite a low grey-level value (almost white, grey level value of 0) and the rest of the material is of a uniform thickness. Thus, segmentation into two classes should render possible cracks or defects in the material. However, the background has to be considered as another class and therefore the segmentation process has to be performed into three classes. As can be seen from Figure 6.8 b, segmentation only into two classes puts the cracks or defects in the same class with the surrounding background. Segmentation into three classes renders the appropriate desired results and it is obvious from Figure 6.8 c that all the cracks are segmented accordingly. After segmentation, the high-level detection process has to take into consideration all possible defects and to take a final decision. The geometrical pre-filtering fuzzy logic module is again not used, since the difference between the average grey-level value of a possible defect and its surrounding background is high. However, the size of an object is important. Therefore, only one fuzzy module can be used that combines the geometrical attributes of the object and its grey-level characteristics. Its input measures are the DIFFERENCE as defined in paragraph (4.5.2.3) and the AREA of the object in question. The system output is (like in other inspection cases) of a bivalent nature: either DEFECT or NORMAL material. The Mamdani fuzzy inference systems consists of the following fuzzy rules:

1. if AREA is SUITABLE and DIFFERENCE is HIGH then OUTPUT is DEFECT
2. if AREA is SUITABLE and DIFFERENCE is LOW then OUTPUT is NORMAL
3. if AREA is NOT-SUITABLE then OUTPUT is DEFECT

The results can be seen in Figure 6.8 d. All cracks were detected successfully. However, the case of detection of cracks in welds is an easy task due to the nature of the defects. Metal has quite a high absorption coefficient as compared to the air/gas and therefore there is a high difference in contrast between cracks and normal material.

6.3.2.2 Inspection of castings and moulds

When mass-produced castings should be 100 percent inspected piece by piece online, the time required to inspect each piece is most important. Inspection for internal defects of casting parts (cavities, cracks, inclusions of foreign material) can be

performed using the proposed inspection X-ray imaging inspection system. In castings or moulds, the most commonly met defects are:

- micro-cavities: spongy, aerated or microporous structure at positions in the casting which are the last to solidify
- surface blow-holes: gases entrapped by solidifying metal on the surface of the casting which result in a rounded or oval blow hole as a cavity

Figure 6.9a shows micro-cavities within aluminium castings. In this particular inspection case, geometrical characteristics of the objects are not important. Although the segmentation into two classes detects some of the porosities within the casting, it cannot segment all possible defects (see figure 6.9 b). Segmentation into 4 classes seemed to generate appropriate results in terms of the number of defects segmented (figure 6.9 c and d). Detection of defects is an easy task due to the uniform thickness of the inspected product or uniform thickness of the parts of the castings. An X-ray image of a defect-free casting comprises only homogenous areas with sharp edges between areas. Therefore, one can detect defects by looking for objects that have different average high-level values than their uniform background. One fuzzy detection of defects module can be used with the following characteristics:

- Input measures:
  - AREA – physical area of the object;
  - BACKGROUND uniformity measure: the object's background can be HOMOGENEOUS or HETEROGENEOUS;
  - DIFFERENCE between the BRIGHTNESS of the object and its surrounding background;
- Output: NORMAL or DEFECT

The fuzzy rules are summarised below:

1. if AREA is SUITABLE and BACKGROUND is HOMOGENOUS and DIFFERENCE is HIGH then OUTPUT is DEFECT
2. if AREA is SUITABLE and BACKGROUND is HETEROGENEOUS and DIFFERENCE is HIGH then OUTPUT is DEFECT
3. if AREA is SUITABLE and BACKGROUND is HETEROGENEOUS and DIFFERENCE is LOW then OUTPUT is NORMAL
4. if AREA is NOT SUITABLE then OUTPUT is NORMAL

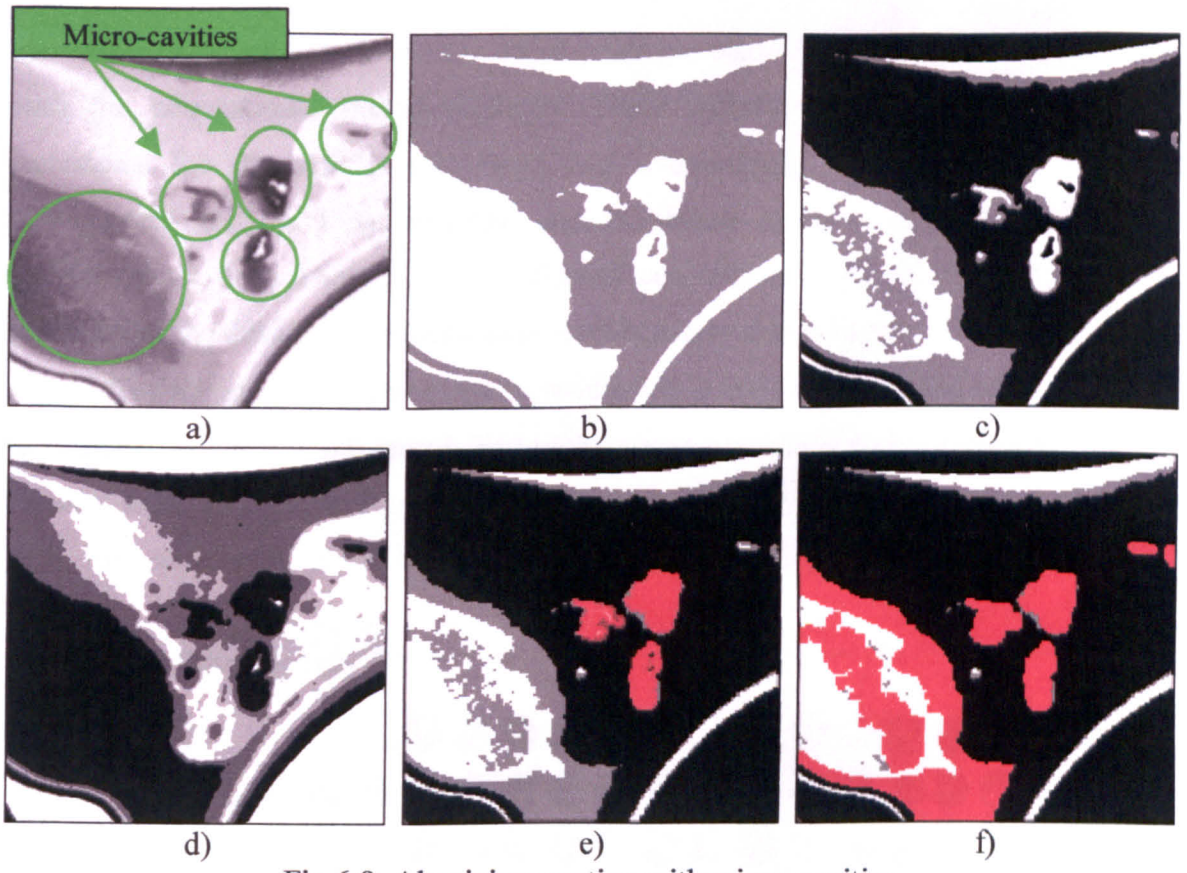


Fig.6.9 Aluminium casting with micro-cavities

a) acquired X-ray image; b) segmented image into 2 classes; c) segmented image into 3 classes; d) segmented image into 4 classes;

e) detection results for low sensitivity (micro-cavities are depicted in red); f) detection results for high sensitivity (micro-cavities are depicted in red)

One can also specify a sensitivity parameter for the detection process. For a low sensitivity only obvious and usually quite small defects are detected (Figure 6.9e and f). In contrast, when using a high sensitivity of the detection process, the size of the object is not important. Thus, this process can render large areas of the product that can be considered faulty areas.

6.3.3 Application 4: Electronics

6.3.3.1 Inspection of Ball Grid Arrays (BGAs)

The BGA package construction is different from conventional leaded packages in several ways. BGA typically uses a natural resin organic substrate (similar to PCB) for die and wire attachment. This substrate incorporates metallized trace routing for connection of the die to the system board through solder balls instead of metal leads

used in leaded packages. However, BGA packages also have several common features with the conventional leaded packages. These include similar chip level assembly techniques for die attach, wire bond and overmolding. BGA packages also use conventional test, packing, handling and PC board assembly methods. Without a method to assure the integrity of bonded interfaces, control of the process and long term reliability of the device at the component level becomes questionable.

Inspection of this component in production is usually done using real time X-ray to evaluate solder ball placement, bad registrations, deformed balls, bad alignment and missing balls and shorts between balls. A typical image showing absent, misaligned balls, balls with voids, and a short between two balls is presented below (Figure 6.10).

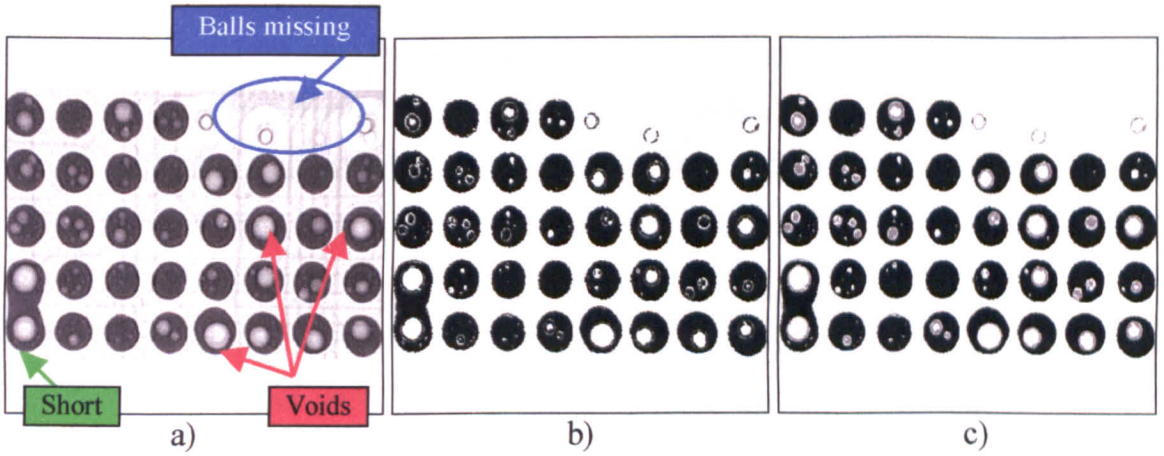


Fig.6.10 X-ray image of a BGA that contains a short, voids and balls missing; b) segmentation result into 2 classes; c) segmentation result into 3 classes

When segmenting such an X-ray image into two classes, only the outlines of the voids present in the balls are detected (Figure 6.10 b). Therefore, segmentation into three classes was performed (Figure 6.10 c).

The high-level detection process is in this case mainly based on geometrical features of the objects extracted from the resulting segmented image. The following features were extracted from the segmented objects:

- ROUNDNESS measure, as defined in paragraph (4.5.2) (HIGH, MEDIUM and LOW)
- AREA , the physical area of the object (SUITABLE, NOT-SUITABLE)
- HOLE, the presence of one or more voids in the object (YES, NO)

- CENTER, the x and y coordinates of the centre of an object (REGISTRED, NOT_REGISTRED); the CENTER measure is computed using the known map of a BGA of the type that one is inspecting.

The OUTPUT of such an inspection system has to be able to recognise all types of defects for one object: VOID, SHORT, DEFORMED, MISALIGNED and NORMAL; if a ball on the grid has all defects present then only one is reported due to the fact that different combinations can lead to a large number of output classes.
The following fuzzy rules can be extracted:

1. if ROUNDNESS is HIGH and AREA is SUITABLE and HOLE is NO and CENTER is REGISTRED then OUTPUT is NORMAL
2. if ROUNDNESS is HIGH and AREA is SUITABLE and HOLE is YES and CENTER is REGISTRED then OUTPUT is VOID
3. if ROUNDNESS is MEDIUM and AREA is SUITABLE and HOLE is NO and CENTER is REGISTRED then OUTPUT is DEFORMED
4. if ROUNDNESS is LOW and AREA is SUITABLE and HOLE is NO and CENTER is REGISTRED then OUTPUT is DEFORMED
5. if ROUNDNESS is HIGH and AREA is NOT_SUITABLE and HOLE is NO and CENTER is REGISTRED then OUTPUT is DEFORMED
6. if ROUNDNESS is HIGH and AREA is SUITABLE and HOLE is NO and CENTER is NOT_REGISTRED then OUTPUT is MISALIGNED

More rules can be derived in order to give the end user more accurate details about the type of ball that it is under investigation. However, for the present argument, 6 rules were used with very good results. All ball with defects were detected accordingly. However, the short present in the X-ray image was not detected. That requires another input measure called SHORTCIRCUIT. It is a feature that is extracted from each segmented object and it is basically checking if two centres (from the known theoretical registration map) are contained within the current object. The values for this crisp input measure can be either YES or NO. Therefore, another fuzzy rule was implemented as follows:

7. if ROUNDNESS is LOW and AREA is NOT_SUITABLE and CENTER is NOT_REGISTRED and SHORTCIRCUIT is YES then OUTPUT is SHORT

Adding the above rule to the above mentioned fuzzy inference system renders all present defects in the inspected X-ray BGA image.

6.3.3.2 Inspection of die-attach

The purpose of die-attach is to provide a reliable attachment of the semiconductor chip to the package while meeting functionality requirements of the package as a whole. The idea behind it is the industry drive for smaller packages that will fit into portable consumer electronics such as cell phones, medical devices, portable computers, pagers, etc. Furthermore, while using faster and powerful chips, the thermal requirements are much more rigorous. The most encountered defect in the die-attach process is the void. Voids that appear between the chip and the package can shorten the life cycle of the chip and can also affect the chip's performance.

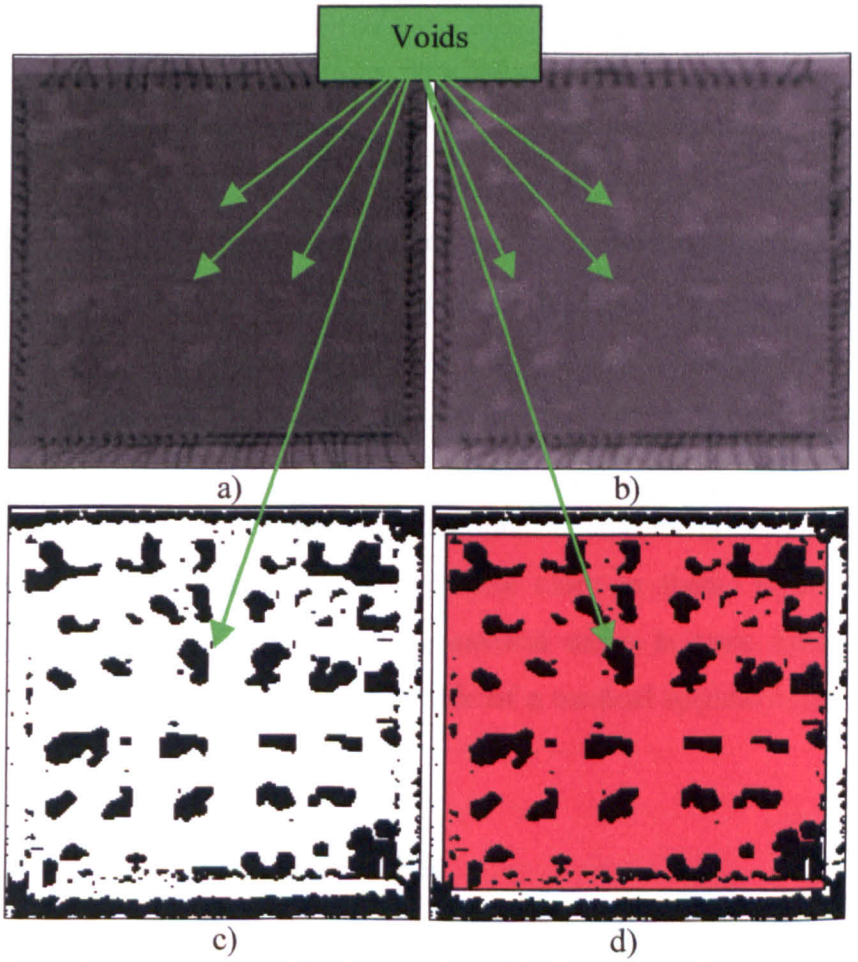


Fig.6.11 a)X-ray image of chip with die-attach voids; b) Image after pre-processing and contrast enhancement; c) segmented image into two classes; d) frame for the computation of the amount of die-attach voids

An inspection system of the die-attach process can provide a method to calculate precisely the amount of die-attach voids. By comparing the results with an accepted amount, one can reject or accept the final product. An example image of a chip with die-attach void is illustrated in Figure 6.11 a. After pre-processing and contrast enhancement, the X-ray image (Figure 6.11 b) is segmented into two classes (Figure 6.11 c). The margins of the chip that contains the connection pins are not taken into consideration when computing the amount of die-attach voids (outside the calculations rectangle depicted in Figure 6.11 d). Thus, the computation is merely a process of counting the black and white pixels, and to compute a corresponding ratio. The resulting ratio is compared with a look-up table containing accepted ratios and a final decision is taken whether the chips is accepted or rejected. As it can be seen, there is no high detection fuzzy logic module necessary here. It is just a simple comparison between two ratio values.

6.4 Classification and diagnostic tasks for medical applications

The proposed system could be used for difficult diagnostic tasks such as in medicine using "fuzzy images". Since even minor errors can lead to catastrophic results, such a system must be able only to assist medical personnel to accurately diagnose patients and not to take the final decisions by itself. False positives and false negatives are equally important in this case. Diseased patients that are not treated (FPs) or healthy patients that are treated for a disease that does not apply to them are not acceptable in the health system. Medical staff will have the final decision when assessing a patient, but the proposed system might provide valuable clues to them. Therefore, how the generic proposed system is applicable to medical cases is very different than the inspection system for the industry. An example of a medical application is presented as follows.

6.4.1 Prediction of Pulmonary Emboli (PE)

Blood clots in the lungs are called pulmonary emboli (PE). This is a potentially fatal condition so a rapid, accurate diagnosis is essential for the correct management of the patient. Treatment commonly involves the use of anticoagulants over a long period of time, which in itself carries some risk. Hence, it is important not to treat patients who do not have the disease *[Innocent et al., 2001]*.

The most common imaging technique for diagnosing PE is the radioisotope lung scan. This technique involves two scans - a perfusion scan to image the blood supply to the lungs and a ventilation scan to image the airways. In order to obtain the images, small amounts of radioactive material are administered to the patient and images of the distribution of the radioactive material are obtained using a specialised radiation detector (the 'gamma camera') *[Innocent et al., 2001]*.

To obtain the diagnosis, these two sets of images are visually compared. PE are demonstrated on the perfusion scan by areas lacking in uptake of radioactive material ('defects') due to reduced perfusion. Unfortunately, although this finding is very sensitive and a normal perfusion scan effectively eliminates PE as the diagnosis, the appearance of perfusion defects is non-specific. This is because areas of reduced uptake can be due to various conditions including infection, fluid in the lungs and chronic respiratory disease. To differentiate perfusion defects due to PE from other causes, a ventilation scan is performed. Nearly all diseases other than recent PE will cause both perfusion and airways abnormalities so perfusion defects will be matched by ventilation defects. Recent PE will cause a change in blood supply but not ventilation that leads to mismatched defects. The reporting of the scans involves assessing the presence of any abnormalities on the two scans and assessing the significance of any differences between them *[Innocent et al., 2001]*.

Since practically a dual-band image is obtained from the lungs (perfusion scan and ventilation scan), the HNN segmentation can be applied as explained in paragraph (4.4.2.2). Information from both images contributes to the segmentation process. The input images are very fuzzy and they look corrupted with impulse noise, therefore the segmentation process is quite difficult. Due to the fuzziness of the input images segmentation into two classes is worthless, rendering a result that does not help a human operator in any way. Thus, segmentation into 3 classes or more will give the operator a better understanding of the image (Figure 6.12). A normal perfusion scan and its corresponding segmented image are presented in Figure 6.12.

In the case of PE, defects from the perfusion scan have to match those on the ventilation scan (Figure 6.13). Therefore, one can segment both images separately (Figure 6.14) and then using a fuzzy logic detection module can "compare" the images to see if the defects are matched and also to determine whether those are blood-clots (the comparing process is here is viewed as more of a fuzzy inference system that

checks how the features from one segmented image relate to features from the other segmented image). Furthermore, by superimposing the two-segmented images and by applying a simple median filter to smooth the resulting image, one can have a more accurate visual aid of the scans (Figure 6.15). Using features from areas of different grey-level in the two-segmented images a fuzzy inference system can be build to take a final decision whether blood clots are present in the lungs. Moreover, by using fuzzy input measures such as the shape of the regions (defined for example as a combination of the ROUNDNESS, AREA, SHAPE and PERIMETER as in paragraph 4.5.2.3) and their surrounding background, the system can decide whether the scans are normal or not. A visual comparison between the segmented image of a normal perfusion scan and the superimposed image can provide the medical operator with sufficient clues to help establish a diagnosis.

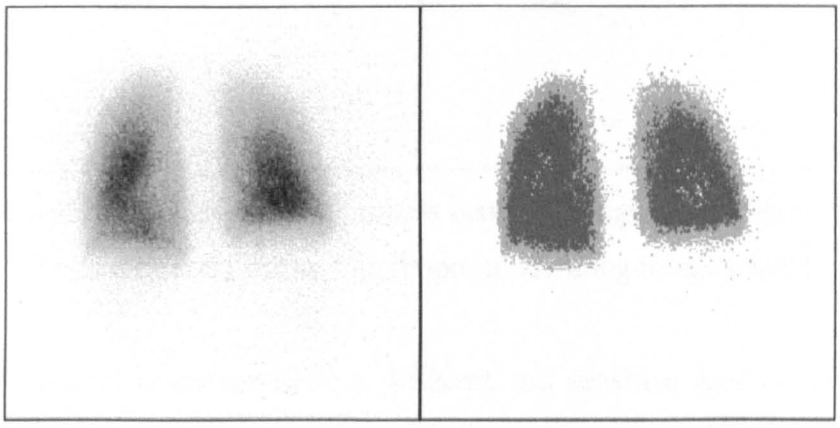


Fig.6.12 Normal perfusion scan and its corresponding segmented image into 3 classes

Fig.6.13 Perfusion and ventilation scan in the case of PE

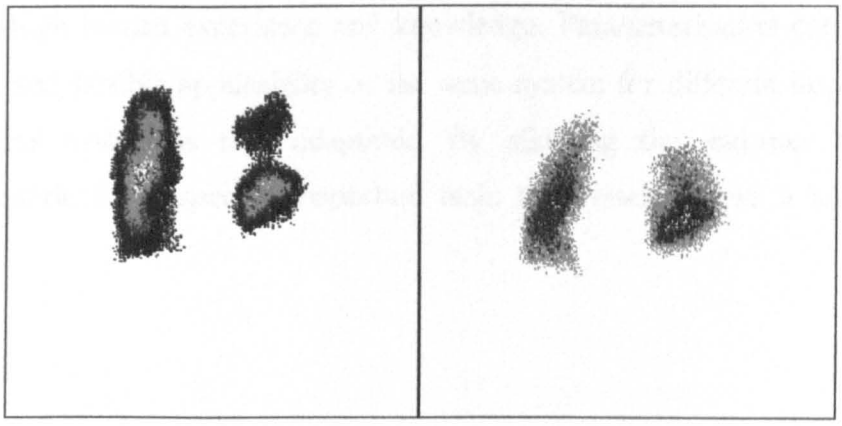


Fig.6.14 Segmentation into three classes of the above images

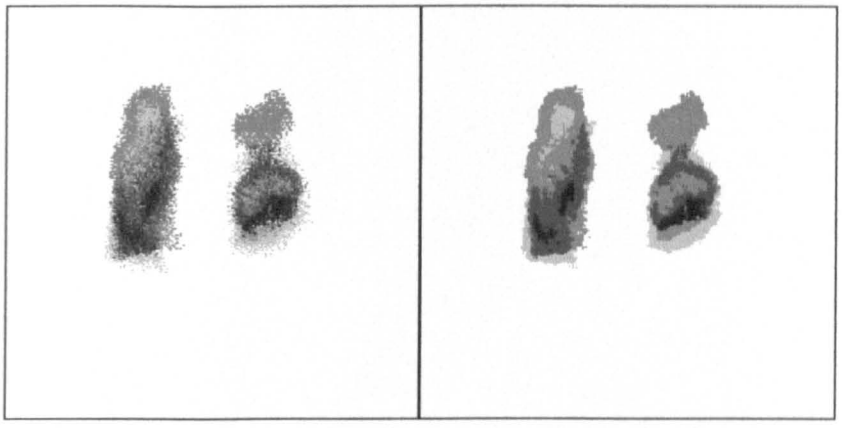


Fig.6.15 Result of the superimpose process between images from Figure 6.14 (left); median filtering of the superimposed resulting image (right)

Since medical cases are quite a different and sensitive area of application and due to lack of medical knowledge and experience, only a proposal of how one can apply the proposed generic imaging diagnostic system is presented. More research is necessary in order to prove the real applicability of such a system. However, it is only a matter of expert advice and parameter tuning in order for such a system to achieve a high performance.

6.5 Summary

The proposed generic X-ray imaging system has successfully been applied in four different applications with very good results. The system can achieve a high robustness in order to meet everyone demands while maintaining a high performance accuracy. This performance is highly dependant on its input parameters that can be

obtained through human experience and knowledge. Parameterisation can lead to fast, easy, robust and flexible applicability of the same system for different inspection tasks. The inspection system is thus adaptable. By allowing the end-user to select its sensitivity criteria for a specific inspection task, the system allows a high-degree of flexibility.

"The best way to escape from a problem is to solve it."

Alan Saporta

C H A P T E R 7

DISCUSSION AND RECOMMENDATIONS

The main conclusions of the present research are presented below. General conclusions are also applicable for the specific context of the inspection of foreign bodies of a meat product in the food industry.

7.1 General discussion

An X-ray imaging inspection system was developed for the application of detection of foreign-bodies embedded within a meat product. This leads to a generic inspection system as described in Chapter 6. It is concluded that this system can be applied to a broad range of fields, from the automotive and aviation industry to medicine. The system was demonstrated to behave in a robust way and incorporates end-user specific inspection options that allows considerable flexibility.

The most important functional modules within the system are a HNN segmentation system and a fuzzy logic high-level detection system. As demonstrated in

Chapter 3 and 4, the segmentation process provides the scaffolding for further image analysis techniques. Particular attention has to be paid to the segmentation process, since its results of later stages are highly dependent on the segmented image. Moreover, special attention has to be paid to the definition of the energy function associated with the HNN. This is domain specific and may vary from application to application. The HNN segmentation algorithm is quasi-automatic in the sense that the only information that needs to be specified *a priori* is the number of segmentation classes. However, that is specified only once, at the design stage of such an X-ray imaging inspection system and depends on the particular inspection task. The conclusion is that HNN provides a sufficient basis for flexibility and robustness in this domain to support the research objectives.

The second important system is the high-level detection system. As explained in Chapter 4, a fuzzy-logic implementation of such a system can be beneficial in terms of performance as compared with other implementations (classical and NN-based architectures). It mainly consists of two modules, one for dealing with geometrical features of the segmented objects and the other one based on grey-level statistical features. Specific inspection characteristics for the desired inspection task obtained from human expert knowledge are incorporated into the above-mentioned modules along with end-user inspection preferences. Thus, this second main system is highly dependant on human experience. Human knowledge translates into fuzzy input measures for the fuzzy-logic modules. However, this constitutes a major advantage for the proposed architecture. It allows the user to choose particular fuzzy input measures, hence conferring on the system the attribute of flexibility. Although this research proposed an architecture with a minimum number of parameters, one may introduce more parameters in order to achieve higher discrimination capacity. This however can increase the computational overhead associated with the algorithms and may slow down the inspection process; thus the real-time analysis capability of the system is reduced. Therefore there is a trade-off between what the end-user desires for their specific inspection task and the actual number of parameters of the proposed system. Furthermore, there is a trade-off between system performance and input parameters as shown in Chapter 4 and 5. The conclusion is that the fuzzy logic processing provides a sufficient basis for flexibility and robustness in this domain to support the research objectives.

In Chapter 5, a ROC curve evaluation was performed. It was demonstrated that the system's performance is comparable to the performance of a human operator. However, the proposed system is not influenced by any human factors such as fatigue and boredom. Hence, the proposed generic inspection system is able to perform at a constant rate of performance and accuracy whereas human operators performance deteriorates over time. The conclusion is that the system can be used in inspection tasks especially where human fatigue is a major problem and a gain in performance will result.

The proposed system architecture based on the use of neural networks and fuzzy inference systems makes it a prime candidate for parallel hardware implementation for real-time application. In this way significant processing speeds can be obtained and practically all-possible end-user requirements may be fulfilled even in the most demanding area of real-time inspection. As mentioned above, the system is highly dependant on the expert knowledge gathered through experience. However, this is done only at the design stage of such an inspection system and not at runtime.

This concludes the section on general conclusions. In the following section, specific conclusions with regards to the inspection process for the food industry are summarised. The general conclusions made above are still valid but not repeated in the following section.

7.2 Specific discussion for the inspection of food products

In the context of the food industry application of the generic X-ray imaging inspection system, the conclusions follow directly from the reported work in the context of finding bones in chicken meat from X-ray image processing. The task of detecting foreign-bodies embedded in meat is a difficult process due to the lack of clarity of the acquired X-ray images. Pieces of meat often overlap creating zones of high contrast that can mask a possible foreign-body present there. However, the proposed system is robust and flexible with regards to its functionality and achieves an acceptable performance from the food industry point of view.

Segmentation with a HNN achieves good results in terms of the correct segmented objects. The corresponding energy function is dependent only on the global distribution of grey-levels present in the input images and no spatial constraints are

introduced. Thus the number of nodes in the HNN architecture is independent upon the image size. Due to the lack of clarity of the images, it was demonstrated in Chapter 4 that using dual-images offers better results as opposed to single-image segmentation. Complementary information from both components of the dual band high and low energy images, was used in the definition of the energy function so that improved segmentation was obtained.

As shown in Chapter 4, possible impulse noise that might corrupt the input X-ray images due to video transmissions via coaxial cables does not affect the performance of the system.

The relatively low computational overhead and the fast speed of convergence make this segmentation algorithm perfect for real-time applications (see considerations in Chapter 4).

The entire system achieves a high performance with a minimum number of parameters. However, choosing the parameters is highly if not entirely dependant on human expert experience.

ROC curve evaluation has proved that the system performance was comparable with human performance. However, human factors such as fatigue do not influence the proposed inspection system and therefore a constant performance over time is achieved.

The conclusion is that the system has been shown to perform very well in the specific context of the food industry with a minimum of development effort involving rule specification and segmentation levels.

7.3 Recommendations and further work

Further work needs to be done in order to improve both the specific system performance and the generalization capability of the generic system.

The pre-processing stage of the images has to be investigated further. Other methods for contrast enhancement when dealing with images with fuzzy edges can render better images as input for the segmentation algorithm. Better improvment of the images leads to better segmentation results, since the results of a certain stage depends on the results of the prior stages in the system.

The applicability of a HNN segmentation method to other types of images needs to be examined further. Segmentation of colour and three-dimensional images using a

HNN can lead to a much wider applicability of the system to, for example, to other image inspection tasks such as infrared or MRI images. Colour images, for instance, can be regarded as a tri-band image comprising three images generated using the red, green and blue colour channels. Applying HNN segmentation is just a matter of redefining the energy associated with the network to incorporate terms from the three component images.

This leads into another issue that can be investigated further: the definition of the energy associated with the neural network. For different applications, the energy definition must comprise terms for specific characteristics of objects on the particular acquired images. In this research the energy was defined solely on the global distributions of grey-levels in the acquired dual-band X-ray images. The energy is dependent of a metric – the distance between two grey levels (defined in the present study as the Euclidian distance, a particular case of the Mahalanobis distance). However, more generalization can be obtained by using a more generalized measure of distance. Furthermore, spatial constraints incorporated into the energy function along with histogram information may lead to improvement of the segmentation process.

The proposed HNN segmentation method requires the number of classes to be known *a priori*; therefore human intervention is necessary during design and development. Further research is necessary in order to automatically generate the number of classes one wants the images to be segmented into.

A solution for escaping the local minimum when trying to minimise the HNN associated energy function such as simulated annealing has to be investigated further. That would lead to better segmentation results, therefore the consequent fuzzy module's performance will be improved.

Most of the inspection system parameters are associated with the fuzzy high-level detection modules. Therefore, the minimization of system parameters can be done at this high level stage of the processing. This means better and more powerful fuzzy measures for the characterisations of inspected objects. More research and analysis by expert human operators can achieve the above stated goal and therefore the minimization of parameters can be achieved. However, since the inspection process is context dependant, more parameters can lead to better discrimination capacity by the system.

Since the acquired X-ray image is highly dependant on the environmental changes such as the intensity of lights or the air temperature in the working area, a self-teaching controller can be used. It should be able to automatically modify the fuzzy sets according to any possible environmental changes, while the system maintains its high performance.

Metrics need to be developed when comparing the performance of two or more architectures. An esential condition is that the comparison should be fair and should not suffer from differences in terms of the input space.

An important area of further work is the code optimisation. Implementation using low level programming languages or machine code will ease the computational overhead associated with the proposed algorithms and speed up the processing. Furthermore, it will enable the design and implementation of massive parallel hardware (Very Large Scale Integrated circuits - VLSI) for the proposed systems algorithms. This will lead to higher processing speeds and consequently to a much more productivity in a wide range of real time applications.

More testing of the generic system in different applications is required in order to prove its high performance, flexibility, robustness and applicability. However, this thesis has demonstrated that a generic system is possible for a wide range of applications.

REFERENCES

Amartur, S.C., Piraino, D., Takefuji, Y., Optimization Neural Networks for the Segmentation of Magnetic Resonance Images, *IEEE Transactions on Medical Imaging*, vol.**11**, no.2, pp.215-220, June 1992

Ambroise, C., Seze, G., Badran, F., Thiria, S., Hierarchical clustering of self-organizing maps for cloud classification, *Neurocomputing*, vol. **30**, pp.47-52, 2000

Amza, C., Graves, M., Innocent, P., Knight, J., Flexible neural network classifier for the automated detection of bones in chicken breast meat, Proc. International Conference on Engineering Applications of Neural Networks 17th - 19th of July 2000, Kingston University, UK, 2000

Amza, C., *Neural network-based classification of bones from X-ray images*, MSc. Dissertation, De Montfort University, Leicester, 1999

Annis, M., Riley G.P., Inspection Method and apparatus with single color pixel imaging, U.S. Patent no. 5,253,283 1993

Asari, K.V., Srikanthan, T., Kumar S., Radhakrishnan, D., A pipelined architecture for image segmentation by adaptive progressive thresholding, *Microprocessors and Microsystems*, vol. **23**, pp 493-499, 1999

Babaguchi, N., Yamada, K., Kise, K., et al., Connectionist model binarization, Proc. 10th ICPR, pp.51-56, 1990

Badono et al., Apparatus for radiation analysis, US Patent no.4,817,122, 1989

Bentazos, A.A., Varela, B.A., Martinez, A.C., Analysis and evaluation of hard and fuzzy clustering segmentation techniques in burned patient images, *Image and Vision Computing*, vol. **18**, pp. 1045-1054, 2000

Berks, G., Keyserlingk, D.G.v, Fuzzy sets in medical image processing, in *Fuzzy Systems in Medicine*, P.S. Szczepaniak, P. Lisboa, J. Kacprzyk (editors), pp. 281-315, Physica-Verlag, Heidelberg, Germany, 2000

Berks, G., Keyserlingk, D.G.v, Segmentation and knowledge presentation in medical image processing using fuzzy sets, in proc. EUFIT'98, pp. 1832-1836, Aachen, Germany, 1998

Bezdek, J.C., Hall, L.O., Clark, M., Goldgof, D., Segmenting medical images with fuzzy models: an update, *Fuzzy Information Engineering*, pp.69-92, John Wiley and Sons, 1997b

Bezdek, J.C., Hall, L.O., Clark, M.C., Goldgof, D.B., Clarke, L.P., Medical image analysis with fuzzy models, *Statistical Methods in Medical Research*, vol.**6**, pp.191-214, 1997a

Bezdek, J.C., Hall, L.O., Clarke, L.P., Review of MR image segmentation techniques using pattern recognition, *Med. Phys.*, vol.**20**, no.4, pp. 1033-1048, Jul/Aug, 1993

Blanz, W.E., Gish S.L., A connectionist classifier architecture applied to image segmentation, Proc. 10th ICPR, pp.272-277, 1990

Bombardier, V., Perez-Oramas, O., Bremont, J., Integrating quality in fuzzy reasoning edge detection, 0-7803-5877-5/00, IEEE, 2000

Cannon, R.L., Dave, R.L., Bezdek, J.C., Efficient implementation of fuzzy c-means clustering algorithms, *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. **8**, no.2, 1986

Chakravarthy, S., Ramamurti, V., Ghosh, J., A network of oscillating neurons for image segmentation, *Intelligent Engineering through ANN*, volume **5**, Dagli, Akay, Chen Fernanded and Ghosh (editors), ASME Press, Nov. 1995

Chanda, B., Kundu, M.K., Padmaja, Y.V., A multi-scale morphologic edge detector, *Pattern Recognition*, vol.**31**, no.10, pp.1469-1478, 1998

Chapanis, A., *Human factors in systems engineering*. New York: John Wiley & Sons; 1996

Chen, C., *Statistical Pattern Recognition*, Hayden, Washington, 1973

Cheng, K.S., Lin, J.S., Mao, C.W., The application of competitive Hopfield Neural Network to medical image segmentation, *IEEE Transactions on Medical Imaging*, vo.**15**, no.4, pp.560-567, August 1996

Cheriet, M., Said, J.N., Suen, C.Y., A recursive thresholding technique for image segmentation, *IEEE Trans. on Image Processing*, vol.**7**, no.6, pp.918-921, June, 1998

Chien, Y.P., Fu, K.S., Preprocessing and feature extraction of picture patterns, TR-EE 74-20, Purdue University, West Lafayette, Indiana, 1974

Cho, S.-M., Cho, J.-H., Thresholding for edge detection using fuzzy reasoning technique, Proc. ICCS, Singapore, vol.3, pp.1121-1124, 1994

Chow, C.K., Kaneko, T., Automatic boundary detection of the left-ventricle from cineangiograms, *Comput. Biomed. Res.*, vol. **5**, pp 388-410, 1972

Clarke, L.P., Velthuizen, R.P., Camacho, M.A., Heine, J.J., Vaidyanathan, M., MRI Segmentation: Methods and Applications, *Magnetic Resonance Imaging*, vol.**13**, no.3, pp.343-368, 1995

Coppini, G., Poli, R., Legitimo, R., De Dominicis, R., Valli, G., A neural network system for detecting lung nodules in chest radiograms, In *Computer Assisted Radiology*, CAR'93, pp. 594-599, Springer-Verlag, Berlin, 1993

Cox, E., *The Fuzzy Systems Handbook*, Academic Press Ltd., London, 1994

Deravi, F., Pal, S.K., Gray level thresholding using second-order statistics, *Pattern Recognition Letters*, vol. **1**, pp. 417-422, 1983

Dumitrescu, D., Hariton, C., *Retele Neuronale – teorie si applicatii* , Ed. Teora, Bucuresti, Romania (in romanian), 1996

Dzubay, Polarization excitation device for X-ray fluorescence analysis, US Patent no.3,944,822, 1976

Eckhorn, R., Reitboeck, H.J., Arndt, M., Dicke, P., Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex, *Neural Computing*, vol.**2**, pp.293-307, 1990

Elsten, S.E., Apparatus for separating meat from poultry bones, US Patent no. 5,064,403, 1991

Engel, M., Neys, J.-L., Ferray, N., Outlines detection by fuzzy logic, in proc. EUFIT'97, pp. 1929-1933, Aachen, Germany, September, 1997

Faruq, A., Closed Pack Contents Inspection without X-rays, *Sensor Rev.*, **11**, 1991

Fletcher-Heath, L.M., Hall, L.O., Goldgof, D.B., Murtagh, F.R., Automatic segmentation of non-enhancing brain tumors in magnetic resonance images, *Artificial Intelligence in Medicine*, **21**, pp. 43-63, 2001

Friddell , System for radiographically inspecting an object using backscattered radiation and related method, US Patent no.4,974,247, 1990

Fu, K.S., Mui, J.K., A survey on image segmentation, *Pattern Recognition*, vol.**13**, pp.3-16, 1981

Fukanage, K., *Introduction to Statistical Pattern Recognition*, New York: Academic, pp.260-267, 1972

Funahasi, K., On the approximate realization of continous mappings by neural networks, *Neural Networks*, vol.**2**, pp. 303-314, 1989

Gaillard, A., Wunsch II, D.C., Escobedo, R.A., Neural hypercolumn architecture for the preprocessing of radiographic weld images, SPIE, vol.1294 Applications of Artificial Neural Networks, pp.378-388, 1990

Ghosh, A., Pal, N.R., Pal, S.K., Image segmentation using a neural network, *Biol. Cybern.*, **66**, pp.151-158, 1991

Glasbey, C.A., An analysis of histogram-based thresholding algorithms, *CVGIP: Graphical Models and Image Processing*, vol.**55**, no.6, pp. 532-537, November, 1993

Gonzalez, R.C., Wintz, P., *Digital Image Processing*, second edition, Addison-Wesley Publishing Company, 1987

Gosh, A., Pal, N.R., Pal, S.K., Image segmentation using a neural network, *Biol. Cybern.* **66**, pp. 151-158, 1991

Graves M. et al., Approaches to foreign body detection in foods, *Trends in Food Science and Technology* vol.**9**, January 1998

Graves, M., *X-ray Machine Vision for on-line Quality Control in Food Processing*, University of Cardiff, Unpublished PhD Thesis, 1999

Gudmundsson, M., El-Kwae, E.A., Kabuka, M.R., Edge detection in medical images using a genetic algorithm, *IEEE Transactions on Medical Imaging*, vol.**17**, no.3, pp.469-474, June 1998

Hall, L.O., Bensaid, A.M., Clarke, L.P., Velthuizen, R.P., Silbiger, M.S., Bezdek, J.C., A comparison of neural networks and fuzzy clustering techniques in segmenting magnetic resonance images of the brain, *IEEE Trans. Neural Network*, vol. **3**, no.5, pp.672-682, 1992

Haralick R.M., Shapiro, L.G., Survey, image segmentation technique, *Comput. Vision Graphics Image Process (CVGIP)*, vol.**29**, pp.100-132, 1985

Haralick, R.M., Shanmugam, K., Dinstein, I., Textural features for image classification, *IEEE Trans. Syst., Man, Cybern.*, vol. **SMC-3**, no.6, pp.610-621, 1973

Haring, S., Viergever, M.A., Kok, J.N., Kohonen networks for multiscale image segmentation, *Image and Vision Computing*, vol.**12**, no.6, pp.339-344, July/August 1994

Haris, K., Efstratiadis, S.N., Maglaveras, N., Pappas, C., Gourassas, J., Louridas, G., Model-base morphological segmentation and labelling of coronary angiograms, *IEEE Transactions on Medical Imaging*, vol.**18**, no.10, pp.1003-1015, 1999

Hartman, K., et al., Layered neural networks with Gaussian hidden units as universal approximations, *Neural Computing*, vol.**2**, pp. 210-215, 1990

Haykin, S., *Neural Networks: A comprehensive foundation*, Macmillan College Publishing Company, Inc., USA, 1994

Hazenbroek, J.E., Revolving poultry thigh deboner, US Patent no. 5,001,812, 1991

Heidke, D.J., Bryce, P.R., Ausf, J., Meat deboning, US Patent no. 5,868,613, 1999

Heiland et al., Automated excision of undesirable material and production of starting material for restructured meat, US Patent no.5,256,102 , 1993

Hiltner, J, Fathi, M., Reush, B., An approach to use linguistic and model-based fuzzy expert knowledge for the analysis of MRT images, *Image and Vision Computing*, vol.**19**, pp.195-206, 2001

Hirano, S., Hata, Y., Fuzzy expert system for foot CT image segmentation, *Image and Vision Computing*, **19**, pp. 207-216, 2001

Hopfield J.J., Neurons with graded response have collective computational properties like those of two-state neurons, *Biophysics: Proc.Natl.Acad.Sci.*, USA, vol.**81**, pp.3088-3092, May, 1984

Hopfield, J.J., Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl.Acad.Sci.*, USA, vol.**79**, pp.2554-2558, April, 1982

Hopfield, J.J., Tank, D.W., "Neural" Computation of Decisions in Optimisation Problems, *Biol.Cybern.*, vol.**52**, pp.141-152, 1985

Hornik, K., et al., Multilayer feedforward networks are universal approximators, *Neural Networks*, vol.**2**, pp.359-366, 1989

Innocent P.R., Belton I., Finlay D.B.L., John R.I. , Type 2 Fuzzy Representations of Lung Scans to Predict Pulmonary Emboli. Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, pp.1902-1907, 2001, ISBN 0-7803-7079-1

Jaulent, MC., Bomdardier, V., Bubel, A., Bremont, J., Application of a fuzzy method for the segmentation of renal angiograms, EUFIT'97, September , Aachen, Germany, pp. 2355-2359, 1997

Jendrysik, F., Eichfeld, H., Graumann, R., Fuzzy segmentation method applied to the extraction of kidney boundaries in medical images, EUFIT'97, pp. 2360-2364, September 8-11, 1997

Jones, B.T., Dillard, D., Sosebee, R.S., Expandable poultry deboner with improved stripper disk, US Patent no. 6,007,417 , 1999

Kamgar-Parsi, B., Gualtieri, J.A., Devaney, J.E., Kamgar-Parsi, B., Clustering with neural networks, *Biol. Cybern.*, **63**, pp. 201-208, 1990

Kapur, J.N., Sahoo, P.K., Wong, A.K.C., A new method for gray-level picture thresholding using the entropy of the histogram, *Computer Vision, Graphics, and Image Processing*, **29**, pp.273-285, 1985

Kaye, R., Crowley, J., Medical-device use safety: Incorporating human factors engineering into risk management, U.S. Food and Drug Administration, July, 2000

Kehoe, A., Parker, G.A., An intelligent knowledge based approach for the automated radiographic inspection of castings, *NDT&E International*, vol. **25**, no.1, pp.23-36, 1992

Kehoe, A., *The detection and evaluation of defects in industrial images*, PhD Thesis, University of Surrey, 1990

Kelstrup, N., Bone detection in Frozen Fish Blocks, Proc. 7th ECNDT, May 1998

Keyserlingk, D.G.v., Pohl, G., Use of fuzzy sets in initial medical image processing, in proc. EUFIT'97, pp. 2350-2354, Aachen, Germany, September, 1997

Kim, T.-H., Cho, T.-H., Moon, Y.S., Park, S.H., Visual inspection system for the classification of solder joints, *Pattern Recognition*, **32**, pp 565-575, 1999

Koch , J. et al., Bone detector, U.S. Patent no. 5,847,382 1998

Kohonen, T., *Self-Organization and Associative Memory*, Third Edition, Springer-Verlag Berlin, 1989

Kosko, B., *Neural networks and fuzzy systems*, Prentice-Hall , Inc., USA, 1992

Koss, J.E., Newman, F.D., Johnson, T.K., Kirch, D.L., Abdominal organ segmentation using texture transforms and a Hopfield neural network, *IEEE Transactions on Medical Imaging*, vol.**18**, no.7, pp.640-648, July 1999

Krug et al., Device and method for inspection of baggage and other objects, US Patent no. 5,490,218, 1996

Kung, S.Y., *Digital Neural Networks*, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1993

Kuntimad, G., Ranganath, H.S., Perfect Image Segmentation Using Pulse Coupled Neural Networks, *IEEE Transactions on Neural Networks*, vol.**10**, no.3, pp/.591-598, May 1999

Kurugollu, F., Birecik, S., Sezgin, M., Sankur, B., Image segmentation based on boundary constraint neural network, In Proc. Third International Workshop on Image and Signal Processing, IWISIP'96, Manchester, UK, pp.353-256, 1996

Kurugollu, F., Sankur, B. – Colour cell image segmentation using pyramidal constraint satisfaction neural network, In IAPR Workshop on Machine Vision Applications – MVA'98, 17-19 November 1998, Makuri, Chiba, Japan, 1998

Kurugollu, F., Sankur, B., Image segmentation based on multi-scan constraint satisfaction neural network, *Pattern Recognition Letters*, no.**20**, pp.1553-1563, 1999

Lawson, S.W., *Defect Detection in Industrial Radiographic and Ultrasonic Image*, (PhD thesis), University of Surrey, Guildford, UK, 1996

Lawson, S.W., Parker, A., Intelligent segmentation of industrial radiographic images using neural networks, Proc. of SPIE, vol.2347, pp.245-255, 1994

Lin, W., Chen-Kuo, E., Chen, C.T. Constraint satisfaction neural networks for image segmentation, *Pattern Recognition*, vol.**25**, no.7, pp.679-693, 1992

Liu, X., Wang, D.L., Range Image Segmentation Using a Relaxation Oscillator Network, *IEEE Transactions on Neural Networks*, vol.**19**, no.3, pp. 564-573, May 1999

Mahalanobis, P.C., On the Generalized Distance in Statistics, National Institute of Science in India, vol.12, pp. 49-55, 1936

McKenna, S.J., et al., Using a two-stage artificial neural network to detect abnormal cervical cells from their frequency domain image, Proc. Of Neural Computing Research and Applications: Part Two Queen's University of Belfast, Northern Ireland, 25-26 June 1992

Metz, C.E., ROC methodology in radiologic imaging, *Investigative Radiology*, vol.**21**, part 9, pp.720-733, 1986

Metz, C.E., Some practical issues of experimental design and data analysis in radiological ROC studies, *Investigative Radiology*, vol.**24**, part 3, pp. 234-244, 1989

Meyn, C., Method and apparatus for examining food products by means of irradiation, U.S. Patent no. 5,026,983 1991

Nadler, M., Smith, E.P., *Pattern recognition engineering*, John Wiley and Sons, USA, 1993

Ocleppo, R., Non-destructive X-ray inspection apparatus for food industry, U.S. Patent no. 6.005,912 1999

Oshio, K., Singh, M., Automatic segmentation of magnetic resonance head images using neural nets, Proc. IEEE Nucl. Sci. Symp. And Medical Imaging Conf., pp.1439-1434, 1990

Otsu, N., A threshold selection method from grey-level histograms, *IEEE Trans. On Systems, Man and Cybernetics*, vol. **SMC-9**, no.1, pp. 62-66, January 1979

Ozkan, M., Dawant, B.M., Maciunas, R.J., Neural-Network-Based Segmentation of Multi-Modal Medical Images: A comparative and Prospective Study, *IEEE Trans. On Medical Imaging*, vol.**12**, no.3, pp.534-544, September 1993

Ozkan, M., Sprenkels, H.G., Dawant, B.M., Multi-spectral resonance image segmentation using neural networks, in Proc. IJCNN 90, vo.1, pp.429-437, San Diego, June 1990

Page et al., Method of and apparatus for sensing the ash content of coal, US Patent no.4,486,894, 1984

Pal, N.R., Pal, S.K., A review on image segmentation techniques, *Pattern Recognition*, vol.**26**, no.9, pp.1277-1294, 1993

Papamarkos, N., Gatos, B., A new approach for multilevel threshold selection, *CVGIP: Graphical Models and Image Processing*, vol.**56**, no.5, pp. 357-370, September, 1994

Papamarkos, N., Strouthopoulos, C., Andreadis, I., Multithresholding of color and gray-level images through a neural network technique, *Image and Vision Computing*, **18**, pp. 213-222, 2000

Papanicolopoulos, C.D., et al., X-ray monitoring system, U.S. Patent no. 5,428,657 1995

Pavlik et al., Method and apparatus for X-ray interrogation of a sample, US Patent no. 3,710,104, 1969

Poli, R., Valli, G., Hopfield neural networks for the optimum segmentation of medical images, *Handbook of Neural Computation*, Oxford University Press, chapter.G5.5, pp.1-10, 1997

Raff, U., Scherzinger, A.L., Vargas, P.F., Simon, J.H., Quantitation of grey matter, white matter, and cerebrospinal fluid from spin-echo magnetic resonance images using an artificial neural network technique, *Med.Phys.*, vol.**21**, no.12, pp.1933-1942, December 1994

Ramsey, J.D., et al., Method and apparatus for the detection of foreign material in food substances, U.S. Patent no. 3,995,164 1976

Ranganath, H.S., Kuntimad, G., Object detection using pulse-coupled neural networks, *IEEE Transactions on Neural Networks*, vol.**10**, no.3, pp. 615-620, May 1999

Reason, J., *Human Error*. Cambridge, Cambridge University Press, 1992

Reddick, W.E., Glass, J.O., Cook, E.N., Elkin, T.D., Deaton, R.J., Automated Segmentation and Classification of Multispectral Magnetic Resonance Images of Brain Using Artificial Neural Networks, *IEEE Transactions on Medical Imaging*, vol.**16**, no.6, pp.911-918, December 1997

Rosenfeld, A., Kak, A.C., *Digital Picture Processing*, Academic Press, New York, 1982

Rout, S., Srivastava, P., Majumdar, J., Multi-modal image segmentation using a modified Hopfield Neural Network, *Pattern Recognition*, vol.**31**, no.6, pp.743-750, 1998

Rumelhart, D. and PDP group., Learning internal representations by error propagation, in *Parallel Distributed Processing*, vol. **1**, MIT Press, Cambridge, MA, pp. 318-362, 1986

Sahoo, P.K., Soltani, S., Wong, A.K.C., Chen, Y.C., A survey of thresholding techniques, *Comput. Vision Graphics Image Process (CVGIP)*, vol.**41**, pp. 233-260, 1988

Seppala, J., Korpisaari, P., Varjonen, V., Lehtokangas, M., Saarinen, J., Character segmentation from X-ray image's ID-label, in Proc. EUFIT'97, September 8-11, 1997

Shareef, N., Wang, D.L., Yagel, R., Segmentation of Medical Images using Legion, *IEEE Transactions on Medical Imaging*, vol.**18**, no.1, pp.74-91, January 1999

Shiranita, K., Hayashi, K., Otsubo, A., Miyajima, T., Takiyama, R.,*Determination of Meat Quality by Image Processing and Neural Network Techniques*, Proc. Of FUZZIEEE2000, Vol. 2, pp516-521, San Antonio, Texas, May 2000

Silverman, R.H., Noetzel, A.S., *Image processing and pattern recognition in ultrasonograms by backpropagation*, Neural Networks, vol.3, pp.593-603, 1990

Smith et al., Apparatus for detecting the presence of hard solid particles in a body of softer solid substance, US Patent no.3,736,583, 1971

Smith et al., Apparatus for inspecting bodies for the presence of hard particles, US Patent no.3,777,886, 1972

Swets, J.A., Pickett, R.M., Whitehead, S.F., Getty, D.J., Schnur, J.A., Swets, J.B., Freeman, B.A., Assessment of diagnostic technologies, *Science*, vol.**205**, no. 4407, pp.753-759, 1979

Takahashi, Y., Shioirl, K., Apparatus for detecting foreign matter with high selectivity and high sensitivity by image processing, U.S. Patent no. 6,023,497, 2000

Terman, D., and Wang, D.L. Global competition and local cooperation in a network of neural oscillators, *Physics D*, vol. **81**, pp.148-176, 1995

Teuber, J., *Digital Image Processing*, Prentice Hall International (UK), 1993

Thomas, J.G., Peters, R.A., Automatic segmentation of ultrasound images using morphological operators, *IEEE Transactions on Medical Imaging*, vol.**10**, no.2, pp.180-185, 1991

Toderean, G., et al., *Retele Neuronale*, Imprimeria Ardealul – Cluj- Romania (in romanian), 1994

Tolias, Y.A., Panas, S.M., A fuzzy rule-based system for applying spatial constraints in fuzzy image clustering, in proc. EUFIT'97, pp. 1921-1924, Aachen, Germany, September, 1997

Valli, G., Poli, R., Cagnoni, S., Coppini, G., Neural Networks and Priori Knowledge Help the Segmentation of Medical Images, *Journal of Computing and Information Technology*- CIT 6, vol.**2**, pp.117-133, 1998

Waidelich, F., Eichfel, H., Graumann, R., Segmentation of medical images using fuzzy technique, in *Fuzzy Systems in Medicine*, P.S. Szczepaniak, P. Lisboa and S. Kacprzyk (editors), Physica Verlag, Heidelberg, Germany, 2000

Wang, D.L., Terman, D., Image segmentation based on oscillatory correlation, *Neural Computing*, vol.**9**, pp.805-836, 1997

Wang, T., Zhuang, X., Xing, X., Robust segmentation of noisy images using neural network model, *Image and Vision Computing*, vol.**10**, pp.233-240, 1992

de Waard, W.P., Neural techniques and postal code detection, *Pattern Recognition Letters*, vol. **15**, pp. 199-205, 1994

Watanabe, S., *Pattern Recognition: Human and Mechanical*, John Wiley&Sons, New York 1985

Weszka, J.S., Rosenfeld, A., Threshold evaluation techniques, *IEEE Trans.Syst.Man Cybern.*, **8**, pp.555-556, 1978

Wolpert, D.H., Stacked generalisation, *Neural Networks*, **5**, pp.241-259, 1992

Wols, F., Poultry thigh deboner with movable stripper, US Patent no. 6,027,404, 2000

Woods, K., Bowyer, K.W., Generating ROC curves for artificial neural networks, *IEEE Trans. on Medical Imaging*, vol.**16**, no.3, June, 1997

Yanowitz, S.D., Bruckstein, A.M., A new method for image segmentation, *Computer Vision Graphics Image Process*, **46**, pp 82-95, 1989

Yin, P.-Y., A fast scheme for optimal thresholding using genetic algorithms, *Signal Processing*, **72**, pp.85-95, 1999

Yoshimura, M., Oe, S., Evolutionary segmentation of texture image using genetic algorithms towards automatic decision of optimum number of segmentation areas, *Pattern Recognition*, **32**, pp.2041-2054, 1999

Young, S.S., Scott, P.D., Nasrabadi, N.M., Object Recognition using Multilayer Hopfield Neural Network, *IEEE Transactions on Image Processing*, vol.**6**, no.3, pp. 357-371, March 1997

Zhang, J., Modestino, J.W., Model-fitting approach to cluster validation with application to stochastic model-based image segmentation, *IEEE Trans. PAMI 12*, pp. 1009-1017, 1990

Zimmermann, H.J., *Fuzzy Set Theory*, 3rd Edition, Kluwer Acad. Publ., Boston, 1996

Zurada, J.M., *Introduction to Artificial Neural Systems*, West Publishing Company, USA, 1992

ONLINE RESOURCES

1. Fraunhofer – ISAR - Intelligentes System zur automatischen Röntgenprüfung, http://www.iis.fhg.de/xrt/system/isar/index_d.html, November 2001
2. Barco X-ray Systems, http://194.7.253.113/bmvs//products_n/X_RAY_SYSTEMS/X_ray.htm, November 2001
3. Cintex UK X-ray systems, http://www.cintex.co.uk/xraysystems.html, November 2001

4. Goringkerr X-ray pipeline system, http://www.goringkerr.com/xray/xrayp.htm. November 2001
5. Heimann Systems Group, http://www.heimannsystems.com/, November 2001
6. Loma Food Inspection Systems, http://www.loma.co.uk/, November 2001
7. http://www.partech.com/pvsc/visions-front.htm, June 2000
8. Safeline Metal detection, http://www.safeline.co.uk/solutions/eol.html, November 2001
9. Spectral Fusion – Bone-Scan online inspection system, http://www.spectralft.com/product/bonescan/bonescan.htm , November 2001

PUBLICATIONS

Amza, C., Graves, M., Innocent, P., Knight, J., Flexible neural network classifier for the automated detection of bones in chicken breast meat, Proc. International Conference on Engineering Applications of Neural Networks 17th - 19th of July 2000, Kingston University, UK, pp.8-15, 2000

Amza, C., Intelligent Classifier of bones within chicken breast meat X-ray images, *Scientific Bulletin*, Series A – *Applied Mathematics and Physics*, vol. **62**, no.2, pp. 83-97, UPB, Bucharest, Romania, 2001

Amza, C., Knight, J.A.G., Innocent, P., Segmentation of dual-band images of x-ray chicken breast using a Competitive Hopfield Neural Network, in Proc. of EANN 2001, Cagliari , pp.297-305, Italy, 16-18 July, 2001

Amza C., Innocent, P., Bones detection from chicken breast meat using a Competitive Hopfield Neural Network and fuzzy filtering, in Proc. of EUSFLAT 2001, pp. 210-213, 5-7 September, Leicester, UK, 2001

Amza, C., Innocent, P., Knight, J.A.G, Intelligent x-ray imaging inspection system with applicability for the food industry, submitted for publication in *The Journal of Pattern Recognition*, December 2001

APPENDIX A

PRINCIPLES OF INDUSTRIAL RADIOGRAPHY

The principle of industrial radiography is identical to that of medical radiography. It is based on the unequal absorption of a beam of x-rays or particles by the different constituents of an object, bringing about the production of radiographic shadows visualized by means of an impression on a radiological film or with the aid of image intensifier visualisation on a monitor of the radiation transmitted across the object.

X-rays are commonly used in many industries: foundries, metal-works, chemistry, the oil industry, mechanical engineering, naval and aeronautic structures, food industry, etc.

A.1. Introduction

The discovery of X-rays by Wilhelm Conrad Roentgen in 1895 has opened a new era of science. Immediately after their discovery, they have started to be used in a widely range of applications from medical to industrial purposes.

X-ray is a type of ionising radiation. X-rays may be defined as electromagnetic radiation produced by deceleration of high-energy electrons and/or by electrons transitions in the inner orbits of atoms.

X-rays are electromagnetic radiation of wavelength range $\lambda \approx 0.01 - 10$ nm. E.g. photon energies $E \approx 120$ eV – 120 keV using the formula:

$$E(keV) = \frac{1.24}{\lambda(nm)} \tag{a.1}$$

William Roentgen in 1896, announced the discovery of X-rays and described many of their properties, among which:

- All substances are, to different degrees transparent to X-rays;
- Photographic plates and films are susceptible to X-rays, providing a valuable mean of recording the effects;
- X-rays are not deflected by a magnetic field;
- X-rays cannot be concentrated by lenses.

A.2. Typical X-ray inspection system

A typical system for industrial inspection is presented in figure A.1.

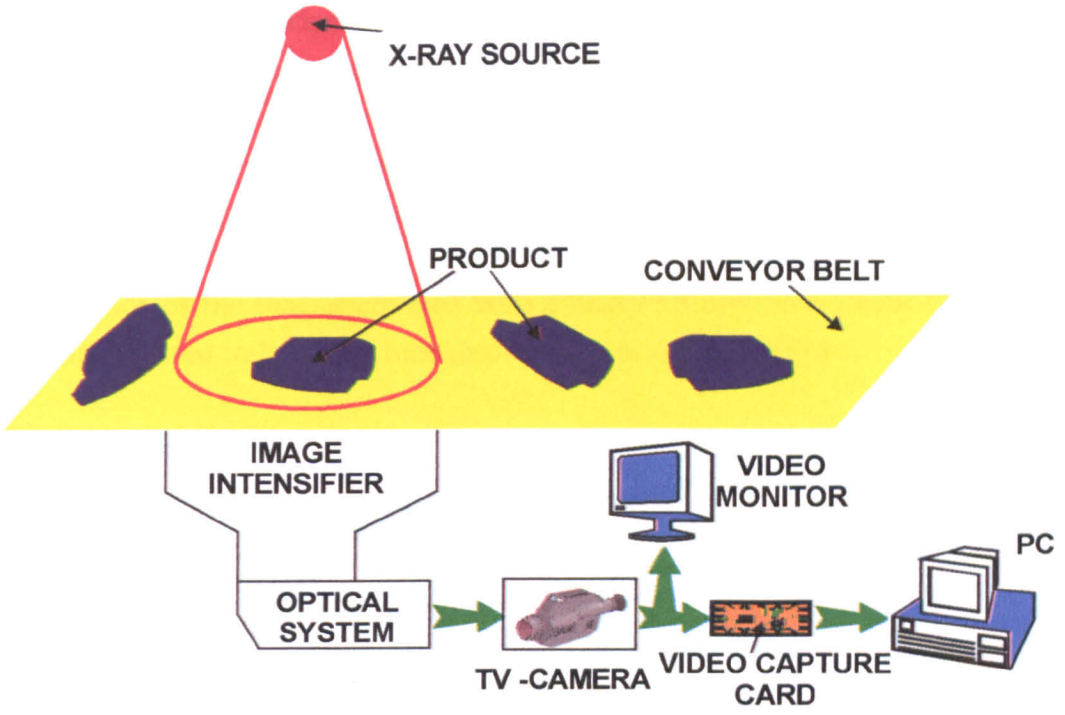


Fig.A.1 System for foreign body detection in meat for the food industry

Such a system comprises an X-ray source, an image intensifier, a TV-camera and a conveyor-belt. The product is passed through the beam emitted by the x-ray source. Below the conveyor belt the image intensifier gathers the remaining x-rays and convert them into a visible image. A TV-camera is used to record that image on-line and to produce a video signal. The video signal can be fed into a video monitor for visual detection or with the aid of a video-capture card fed into a computer for automatic post-processing and detection.

Conventional X-radiography systems use an X-ray generator whose maximal emission voltage is between 20 and 400 kV.

The most widely applied source of x-rays is the x-ray tube, which has many forms for various applications.

The main components of such a system are described in details in the following paragraphs.

A.2.1 X-ray source

The x-rays are produced when high-speed electrons are suddenly stopped by a solid object. So, in order to generate x-rays there should be a source of electrons, a means of accelerating them and a target to stop them.

The most used source of x-rays is the x-ray tube. The most common structure used in X-ray tubes is shown in figure A.2 [*after Whiston, 1987*].

The functioning of an x-ray tube is very simple. The electrons are produced by an electrically heated tungsten filament. In order to accelerate the electrons and thus achieves high speeds, a large potential difference is applied between the filament and the target (e.g. 10-70 kV). The tube is evacuated to create a high vacuum in order to avoid unwanted electrical phenomena between electrons and atoms in the gas. When the high-speed electron hit the target, the atoms of it stop them and X-rays are produced in all directions from the surface of the target. Because the X-rays are very harmful to life, the interior of the tube is covered with a heavy absorption material (e.g. lead). The x-rays are allowed to leave the tube through one or more windows.

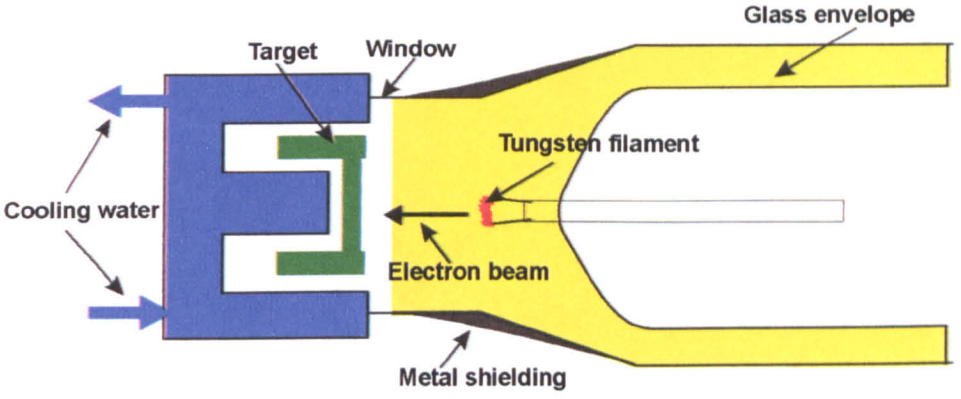


Fig.A.2 X-ray tube

The process of x-ray production is made up of two major components: the production of continuous spectrum of x-ray energies from electrons interacting with atomic nuclei (*Bremsstrahlung*) and the production of characteristic x-rays by the rearrangement of the orbital electrons of a target atom after the ejection of one of them by a high-speed electron (see figure A.3).

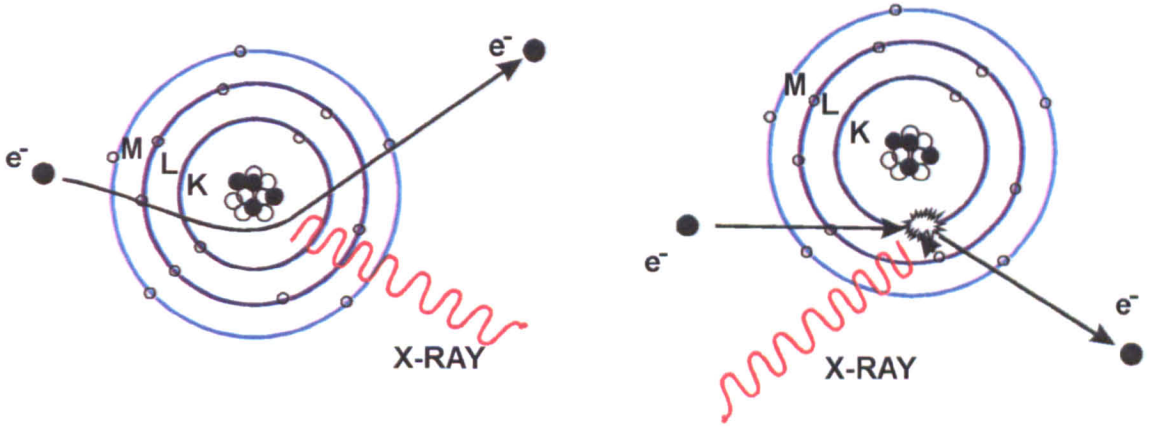


Fig.A.3 X-ray production (general emission – left; characteristic radiation - right)

When a high-speed electron passes close by a positively charged nucleus, its energy is diminuated and its direction changed (attraction to the intense electric field).

This energy will be irradiated as x-rays (*Bremsstrahlung*). Because an incident electron from the tungsten filament can lose any amount of its initial energy, the x-rays produced are not of a specific energy, but are distributed in a form of a continuous spectrum (see figure A.4). Maximum x-ray energy is produced when an incident electron collides with a nucleus.

If an incident electron collides with an orbital electron in the target atoms, and if its energy is sufficient enough to attenuate the binding energy of the specific electron shell, the orbital electron is ejected. The orbital electron leaves a free place in the specific shell. This place will be occupied by another electron from any of the outer shells and so on. These transitions of electrons from orbital shells results in radiation (characteristic x-rays) with energy equal to the difference in binding energies of the two shells.

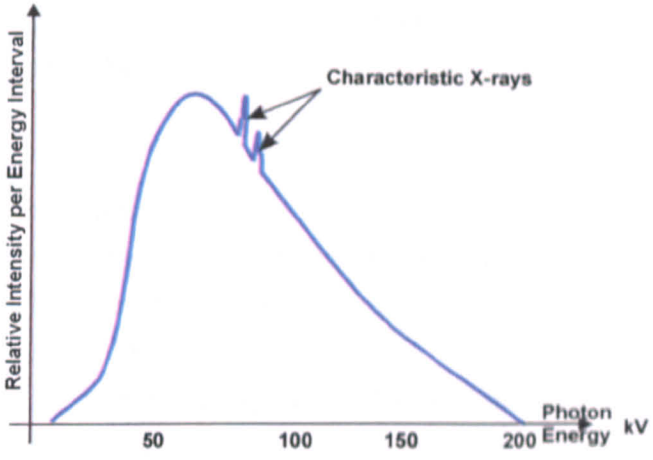


Fig.A.4 X-ray continuous spectrum

The percentage of the total x-ray, contributed by characteristic x-ray, is a function of the high-voltage applied to the x-ray tube. In the region of 50 to 100 kV the characteristics x-ray make up about 25% of the total x-ray produced. When dealing with higher energies (>150 kV), their contribution to the total is almost insignificant.

From the total energy supplied to the tube, only a very small fraction is converted to X-rays (1-5%). Most of the energy is transferred into heat. A cooling system is used to prevent the target from melting. Usually the cooling system consists of water pumped into the target part or the x-ray tube.

The target material has to have a high melting temperature with a good thermal conductivity. Such metals used as typical target material are Mo, W, Cu, Cr (metals with high atomic number for production of high-intensity x-rays).

The number of electrons extracted from the cathode represents the tube current and is measure in milliamperes (mA). The voltage applied between the cathode and the anode determines the energy or speed of the electrons and is measured in kilovolts (kV). The cathode has an important component – the focusing cup. The role of the focusing

cup is to narrow down the diameter of the electron beam so it can be directed only on a small part of the anode, called the focal spot. The size of the filament determines the size of the electron beam, the size of the target on the anode (focal spot), and therefore the size of the base of the x-ray beam. The focal spot size has a substantial effect on the image quality: the smaller the focal size, the higher the image detail is.

Apart from the tube itself, an X-ray generator has a number of other components. These include a high power voltage source, safety devices such as automatic cut-offs if the water supply fails (flow switch), or a fault develops in the tube.

Thus, the most important criteria that differentiate between x-ray machines are:

- The maximum mA of the x-ray unit represents the power of the machine. How high in mA the tube can go determines the amount of x-ray the tube can produce in a given time and therefore will determine the capability of the x-ray unit.
- The focal spot size determines the image detail. The smaller the focal size is, the finer the image will be.

A.2.2 X-ray Image Intensifier

There are a few methods of studying the varying intensities of X-rays after they traversed the product (after some partially and differential absorption of the source beam by some parts in the object to be inspected). One is to record a two-dimensional image on an X-ray film during the exposure. Another method is to visualize a moving shadow on a fluorescent screen. But this method produces a luminance of only 10^{-5}-10^{-2} cd/m^2, not enough to give a reasonable cone vision. This can be overcome by increasing the brightness of at least 1000 times, as stated by Chamberlain in 1942 *[Airth, 1980]*. From here comes the idea of image intensifier. The first commercially available X-ray image intensifier appeared in 1953, but its internal structure is practically unchanged over the time.

The principle of operation of an X-ray image intensifier tube is illustrated in the figure A.5 *[after Schagen, 1980]*. An X-ray beam absorbed in the fluorescent input screen produces several thousands visible photons. These photons are converted into photoelectrons by a photo emissive layer deposited on the inside of the input phosphor screen. This layer will emit one photoelectron for every 5-10 incident photons. The photoelectrons are then accelerated and focused to form an image on the output screen.

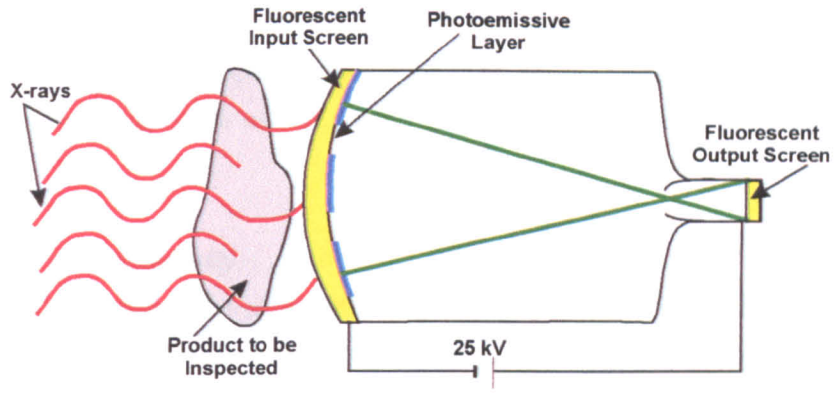


Fig.A.5 Basic X-ray Image Intensifier

A.2.2.1 Image Intensifier characteristics

The most important characteristics for an image intensifier system are described below:

- *Brightness gain*

The brightness gain is the product between the minification gain and the flux gain

$$Brightness\ Gain = (Minification\ Gain)\ x\ (Flux\ Gain) \tag{a.2}$$

The minification gain results because electrons from a large input window are focussed down to the smaller output window. This will lead to an increase in the number of electrons/mm^2. The minification gain is given by:

$$Minification\_Gain = \frac{(Diameter\_of\_Input\_Window)^2}{(Diameter\_of\_Output\_Window)^2} \tag{a.3}$$

The flux gain results from the acceleration given to the electrons, as they are attracted from the input window to the output window. It is typically between 50 and 100, but it is dependent on the applied voltage.

- *Conversion factor*

The conversion factor relates what the image intensifier delivers (i.e. luminance) relative to the input (i.e. radiation exposure).

- *Contrast ratio*

The contrast ratio is given by the formula:

$$Contrast\_ratio = \frac{Luminance\_at\_Center\_of\_Open\_Field}{Luminance\_with\_10\%\_of\_Open\_Field\_Blocked\_by\_Pb\_Disk} \tag{A.4}$$

It can be measured by imaging a lead disk and expressing the luminance of its image relative to that of an open field image. For standardization purposes, the size of the lead disk is typically 10% of the field size. Typical values for this ratio are 20:1 to 30:1.

➢ *Spatial non-uniformity*

Intensified images of a uniform object are generally brighter in the centre than in the periphery due to an unequal brightness gain in different regions of the field of view. This effect is also called ***vignetting*** (fig. A.6).

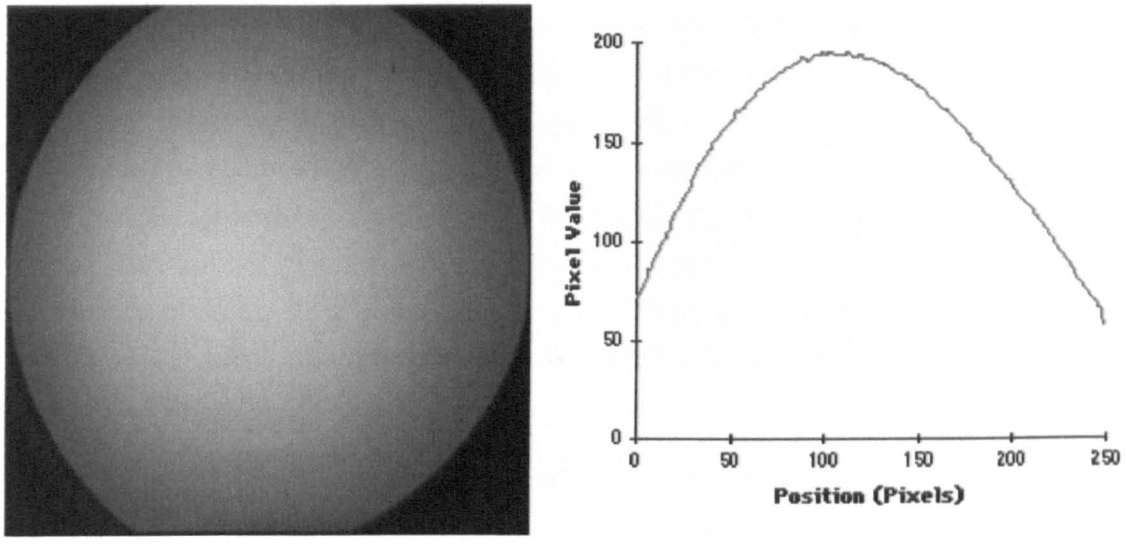


Fig.A.6 Vignetting effect for X-ray image intensifiers

➢ *Spatial distortion*

All image-intensified images suffer from this effect. The images do not reproduce the exact spatial relationships in an object because of unequal magnification in different regions of the field of view (fig.A.7).

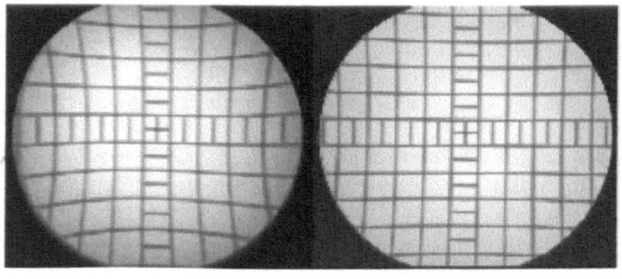


Fig.A.7 "Pincushion" effect for X-ray image intensifiers

The distortion is typically 'pincushion' as it can clearly be seen from the above figure.

'Spiral' distortion (S-distortion) can also appear as a result of an image intensification process. This effect is caused by the magnetic field surrounding the image intensifier.

A.3. Interaction of x-rays with matter

A.3.1 Attenuation of an X-ray beam by matter

The interaction of X-rays can only be explained by assuming the corpuscular nature of this radiation. The wave nature of it only clarifies and explains a few of the phenomena involved in the change of energy between the x-ray radiation and matter.
An incident x-ray beam impinging an absorber is attenuated. This attenuation consists in two processes: absorption of x-ray photons into matter and scattering of photons out of the beam. When we discuss about the processes involved in the loss of energy of the incident x-ray beam when traversing matter we assume that the beam is composed of small particles (photons) and the matter is composed by discrete particles.

The number of interactions occurring between the photons in an x-ray beam and the matter particles in a given thickness of absorbing material is proportional to the number of incident photons upon the absorber *[Ter-pogossian, 1969]*. This can be expressed as:

$$\frac{\Delta N}{N} = -\mu \Delta x \tag{a.5}$$

where ΔN is the number of photons interacting, ΔN is the number of incident photons, x is the absorber thickness and μ is a constant.

The total attenuation of an x-ray beam by an absorber of finite thickness x can be calculated by integrating the expression $dn/dx=-\mu N$ as follows:

$$\int_{N_0}^{N} \frac{dN}{N} = -\mu \int_{0}^{x} dx \tag{a.6}$$

The result of the integration is

$$N = N_0 e^{-\mu x} \tag{a.7}$$

This equation shows the relationship between the attenuation of an x-ray beam by an absorber of thickness x and the linear attenuation coefficient μ for the radiation and the material under consideration.

A.3.2 Interactions between X-rays and particles of matter

The interactions of x-rays with matter take place at a sub molecular level. X-rays interact only with atoms, electrons and nuclei. In any problem involving the interaction of x-rays with matter it is sufficient to specify the atomic mixture of the form of matter

studied, disregarding its molecular composition (the attenuation of a x-ray beam is the same whether the absorber is formed by a mixture of atoms or the same atoms are bound to form some kind of chemical compound).

In general, x-ray photons interact with particles of matter at a scale that is determined both by the size of particle and the energy of the x-ray photon. Low-energy x-ray photons may interact with an atom as a whole and high-energy x-ray photons are more likely to interact with tightly bound electrons in inner electronic shells; the very high-energy x-ray photon interact with an atom at a nuclear level.

X-rays interact by elastic and inelastic scattering and by photoelectric absorption. All of those phenomena are dependent of the x-ray wavelength and the type of absorber (material).

A.3.3 Elastic scattering

Elastic scattering consists of Thomson scattering and Rayleigh scattering.

A.3.3.1 Thomson scattering

Atomic electrons can be tightly or loosely bound. Some of the loosely bound electrons can be considered free because of the weak forces involved.
Thomson scattering occurs from the interaction between a plane electromagnetic wave and a free electron.

This scattering is also referred as classical scattering because it can be completely explained on the basis of the classical theory of electromagnetic radiation. In this way, the interaction consists of a transfer of energy from the x-ray wave to the atom. Thus, the atom is set into vibration and re-emits electromagnetic radiation in phase and equal in energy with the incident radiation. This is why this kind of scattering is called coherent. The new electromagnetic radiation is deflected from the original direction.

Thus, the interaction of an x-ray photon with a whole atom results in the deflection of the direction of travel of the photon, or scattering of the photon from its original direction, without loss of energy to the atom (fig.A.8)[*Ter-pogossian, 1969*].

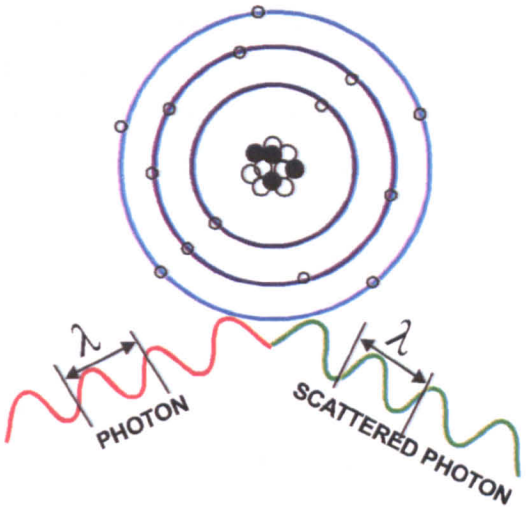


Fig.A.8 Thomson scattering

Coherent scattering can only take place if the wavelength of the electromagnetic radiation is of the same magnitude as the atomic dimensions. Because the diameter of an atom is about 10^{-8} cm, this classical scattering only takes place when the energy of incident x-ray photons is only a few keV. Thus, this scattering is not important from the foreign body detection in meat, where we are dealing with tens of keV (25-150 keV).

A.3.3.2 Photoelectric effect

This is an interaction between an incident x-ray photon and an electron tightly bound in an atom. This phenomenon is accompanied by the emission of an x-ray photon characteristic of the absorber (fig.A.9).

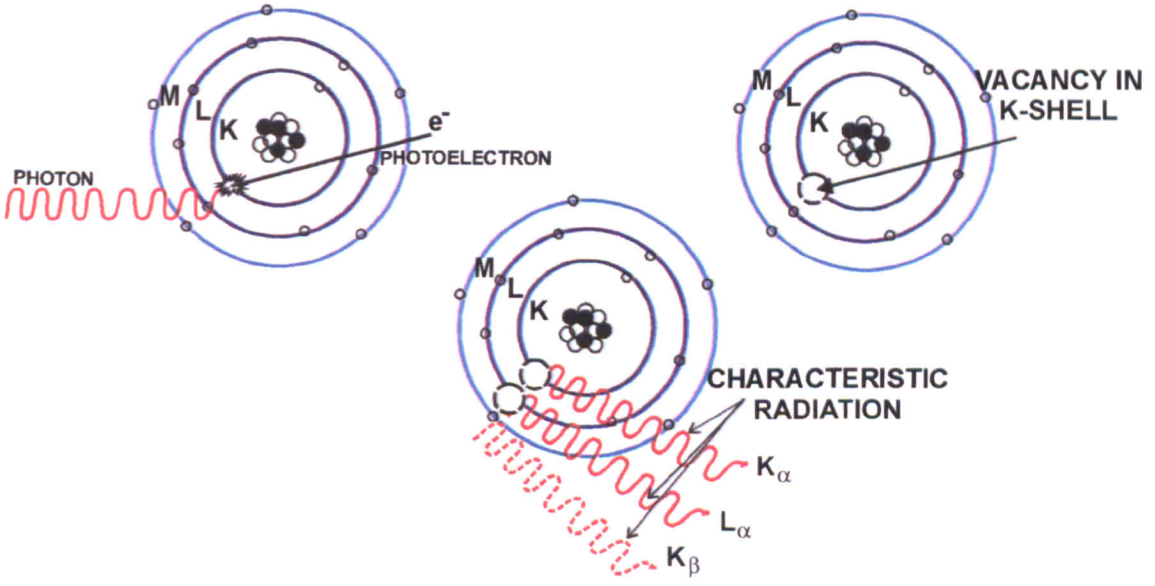


Fig.A.9 Photoelectric effect

The energy of the incident x-ray photon is transferred to the electron, thus the photon ceases to exist. The energy transferred is hn, where h is Planck's constant and n is the frequency of the x-radiation. This energy is divided into two parts. One part, W_0 is used by the electron to overcome the binding energy in order to free itself from the shell. The remaining energy E_{kin} is carried away by the electron in the form of kinetic energy. The transfer of energy can be described by the equation:

$$hn = E_{kin} + W_0 \qquad \text{(a.8)}$$

The kinetic energy is later dissipated in matter, away from the atom, when the photoelectric effect takes place.

The photoelectric effect can take place only if the energy of the incident x-ray photon is high enough to overcome the binding energy for that specific shell. This will result in the creation of a vacancy in the shell under consideration. The vacancy will be shortly filled by an electron from a neighbour shell. In doing so, another vacancy is created in the next shell and the process continues like that. It is actually a cascade-type process of replenishment of electronic shells. This process is accompanied by emission of the x-radiation characteristic (sometimes called fluorescent radiation) of the absorbing medium. If the atomic number of the absorber is low, then the energy of the characteristic energy is also low and is usually absorbed in the immediate vicinity of the absorbing atom. In the opposite case, if the atomic number is high, then these photons can travel long distances away from the absorbing atom.

A.3.4 Compton effect

Inelastic, or Compton scattering occurs from loosely bound electrons and also involves a transfer of energy from the incident x-ray photon. It consists of the deflection or scattering of the incident photon from its original direction, with loss of some energy that is gained by the recoil electron (fig.A.10).

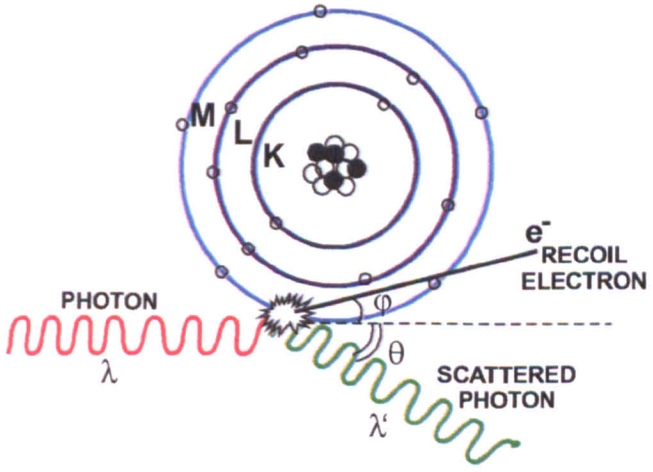


Fig.A.10 Compton scattering

The equations relating to variables in a Compton interaction are *[Ter-pogossian, 1969]*:

- Change in wavelength for the incident photon

$$\lambda - \lambda' = \Delta\lambda = 0.0241(1 - \cos\theta) \ [A^0] \qquad (a.9)$$

where λ is the wavelength of the incident x-ray photon and λ' is the wavelength of scattered photon and θ is the angle of scattering.

- Relation between the angle of scattering (θ) and the angle of the recoil electron (φ)

$$\cos\theta = (1 + \frac{hn}{0.511})\tan\frac{\varphi}{2} \qquad (a.10)$$

where *hn* is the energy of the incident x-ray photon.

If the wavelength of the incident photon is long compared to $0.0482A^0$ (*hn*<50 keV), the maximum change in wavelength represents only a small percentage of the energy carried by the photon, thus the recoiling electron receiving littler energy. In this case, the scattered radiation differs only slightly from the incident photon. On the other hand, with incident photons of high energy (*hn*>100 keV), a large percentage of the incident photon energy is transferred to the recoiling electron.

The Compton effect depends only on the number of electrons in the absorber. Since the number of electrons per atom is equal to the atomic number, this effect (per atom) is proportional to its atomic number.

A.3.5 Interaction between the x-ray photons with nuclei

X-ray photon energy may be exchanged into matter in a different way than the ones already described above. This mechanism is only for high-energy photons and the interaction is with the atomic nucleus and does not lead to a rejection of electrons from the atomic shells. Photons with energies greater than 1.02 MeV may interact with the electric force field of the nucleus. The energy of the photon is converted into two electrons – one negatively charged and the other positively charged (fig.A.11). After the interaction, the photon ceases to exist. The electron and the positron both give up their kinetic energy to the medium by ionising and exciting other atoms. When the positron comes to rest, it combines with one of the free electrons annihilating each other. This phenomenon is also called pair production.

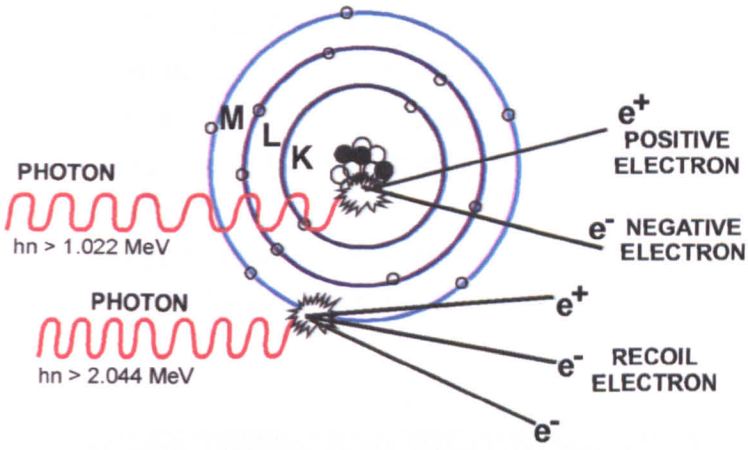


Fig.A.11 Pair and triplet production

The conversion of photon energy into matter may also occur in the vicinity of an electron. This is called triplet production, because the electron is also set in motion along with the other two electrons formed in this process (fig.A.11).
Both pair and triplet production are high-energy processes that play no role in present-day inspection process.

A.3.6 X-ray interaction with meat

The phenomena taking place when an x-ray beam is impinging upon a product meat which contains a bone in it is summarised in the following figure A.12 [*Papanicolopoulos, 1995*].

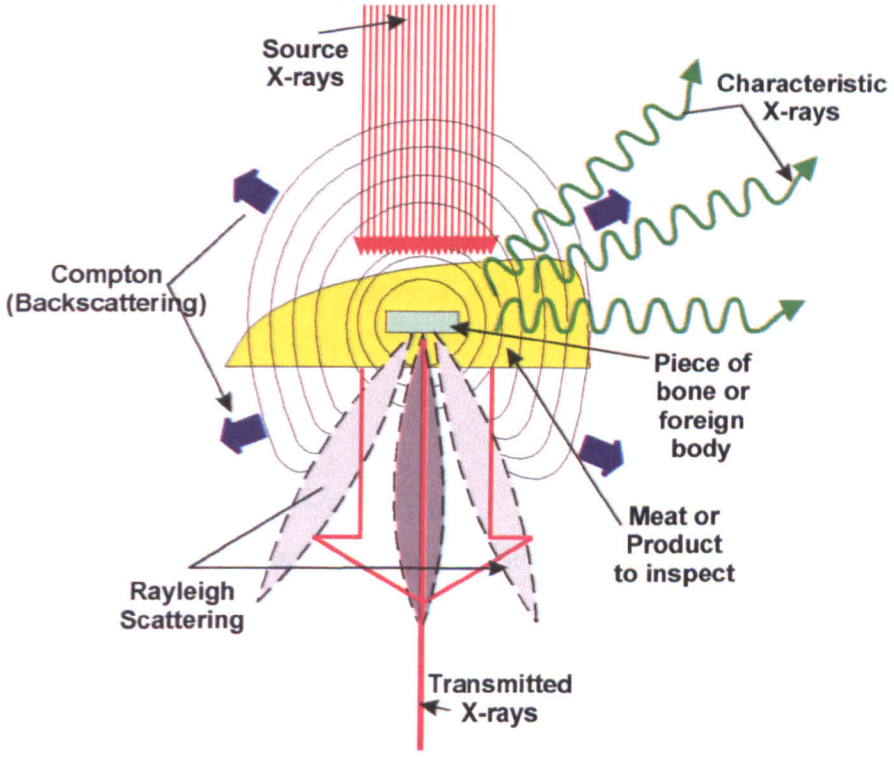


Fig.A.12 X-ray interaction with a piece of meat containing a bone

Although five modes of x-ray attenuation were discussed in this chapter, only photoelectric and Compton interactions are important from the X-ray inspection process point of view. Thomson scattering occurs at energies too low and pair and triplet production occurs at energies too high to be of any consequence in the foreign body detection by means of irradiation.

Predominance of one or more of these processes will depend upon the energy of the incident x-ray photons and the nature of the absorber (table 1, after Pizzarello 1967).

Table A.1

Absorption process	Predominant absorption
Photoelectric	Bone will absorb 5 to 6 times as much energy per gram as will soft tissue
Compton	Bone and soft tissue absorb essentially the same energy per gram
Pair and triplet production	Bone absorbs approximately two times as much energy per gram as soft tissue

At lower energies (<60 keV), the absorption of x-rays is primarily by the photoelectric effect. The photoelectric effect depends on the atomic number of the absorbing medium. Thus, the absorption process is much high for bones than for muscle or air (because of the higher effective atomic number of bone). As the photon energy rises, the Compton process becomes more important. Compton scattering depends only on the number of electrons/gram of the absorbing medium (it is almost the same for air, muscle and bone). Thus, between 60 keV and 200 keV bone rapidly becomes less preferential in absorption.

Because the systems used in the experiments uses low energy x-rays (25kV-65kV), only photoelectric and Compton interactions are taken into account.

If I_0 is the intensity of the incident x-ray beam, then the intensity of the beam I when it reaches the image intensifier system can be expressed:

$$I = I_0 e^{-(\mu_{meat} + \mu_{bone})} \tag{a.11}$$

when the incident beam hits a bone;

and

$$I' = I_0 e^{-\mu_{meat}} \tag{a.12}$$

when the incident beam does not hit a bone, where μ_{meat} and μ_{bone} are the attenuation coefficients for meat, and bone respectively.

In the above equations, the fat, water and skin were not taken into consideration. A complete equation for the attenuation of the intensity of an incident X-ray when traversing a meat product can be written as:

$$I' = I_0 e^{-(\mu_{meat}+\mu_{bone}+\mu_{fat}+\mu_{skin}+\mu_{water})} \quad \text{(a.13)}$$

As far as the attenuation of the number of incident photons, equation (7) can be re-written:

$$N = N_0 e^{-\mu_{total} x} \quad \text{(a.14)}$$

only if the thickness of the product is uniform. This is not the case for the food industry, where the meat has a random thickness.

Experimental values are presented in the following table A.2 [after *Webb et al., 1988*]. The attenuation was measured for fat tissues, muscle tissues and bones, at different photon energies (10-150 keV). The attenuation curve displayed in figure A.13 shows that the greatest attenuation is in the bones and the smallest attenuation is in the fat tissues. This attenuation will lead to a different level of greys in the final X-ray image. The image has to have a good contrast in order to distinguish between different types of tissues. Figure A.14, shows that a good contrast in this case is achieved by small photon energies (i.e. between 10 and 60 keV). When the incident photon energy increases, the attenuation decreases dramatically and the bone, muscle and fat tissues tend to have the same attenuation. Thus, poor contrast is achieved when using high energy x-rays (>60 keV).

Table A.2 – Attenuation of X-ray energy within different types of absorber

	Photon energy [keV]						
Attenuation [cm^2/g]	10	20	30	40	50	100	150
Attenuation in BONE	8	2.7	0.9	0.5	0.35	0.25	0.2
Attenuation in MUSCLE	5	0.9	0.4	0.32	0.25	0.22	0.19
Attenuation in FAT	4.5	0.52	0.3	0.28	0.22	0.21	0.18

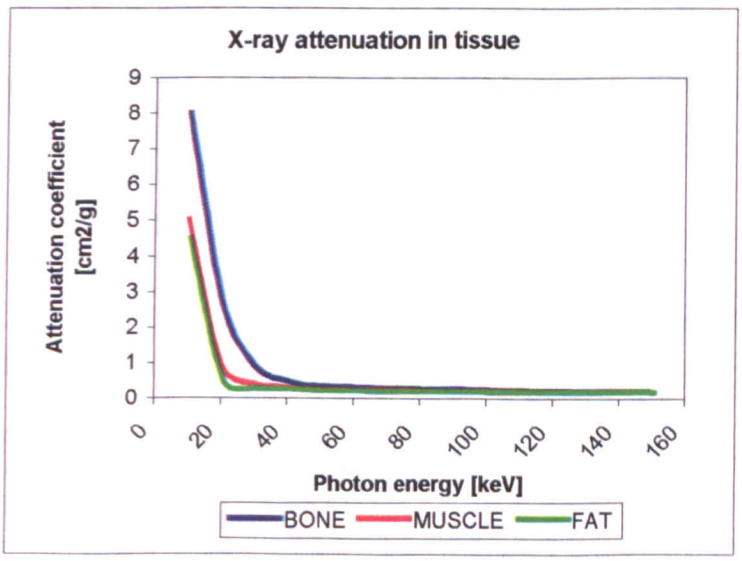


Fig.A.13 X-ray attenuation in different types of tissues

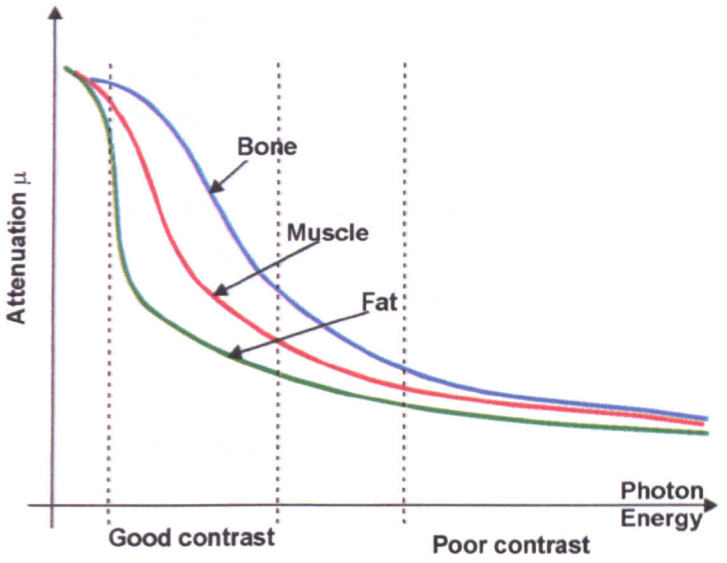


Fig.A.14 X-ray and tissues contrast

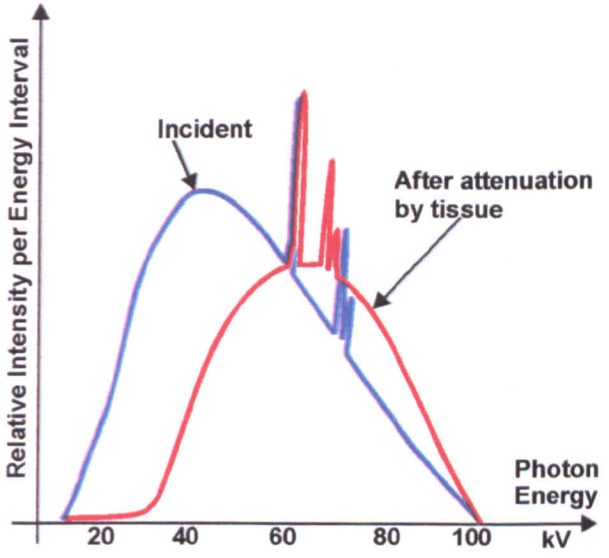


Fig.A.15 X-ray spectrum before and after attenuation by tissues

Figure A.15 shows how the X-ray spectrum is affected after the attenuation.

BIBLIOGRAPHY AND REFERENCES

1. Airth, G.R., *X-ray image intensifiers: applications and current limitations*, in Ultrasound and X-rays in Engineering and Medicine, Royal Society, London, 1979
2. Baba, R., et al., *Method for generating X-ray image and apparatus therefor*, US Patent 5878108, 1999
3. Bertin, E.P., *Introduction to X-ray spectrometric analysis*, Plenum Press, New York, USA, 1978

4. Chen ,Sow-Hsin , Yip, Sidney, *Spectroscopy in biology and chemistry – Neutron, X-ray, Laser*, Academic Press inc., London, 1974
5. Dyson, N.A., *Radiation Physics with applications in medicine and biology*, second edition, Ellis Horwood, Chicester, West Sussex, UK, 1993
6. Granier, R., Gambini, D., *Applied Radiobiology and radiation protection*, Ellis Horwood ltd., Chicester, West Sussex, UK, 1990
7. De Groot, P.M., *Image intensifier design and specification*, Proc. Of the 1991 AAPM Summer School – Specification, Acceptance Testing and Quality Control of Diagnostic X-Ray Imaging Equipment, Vol. I, New York, 1991
8. Hashim, I.B., et al., *Consumer acceptance of irradiated poultry*, Poultry Science 1995, aug., vol.74, pp. 1287-1294
9. Katzenstein, L., *Food irradiation: the story behind the scare*, American Health 1992, vol. 11, n.10, pp.60-64
10. Michette, A.G., Buckley, C.J., *X-ray science and technology*, IOP Publishing, London, 1993
11. Ocleppo, R., *Non-destructive X-ray inspection apparatus for food industry*, US Patent 6005912, 1999
12. Papanicolopoulos, C.D., et al., *X-ray monitoring System*, US Patent 5428657, 1995
13. Pizzarello, D., Witcofski, R., *Basic Radiation Radiology*, Lea and Febiger, USA, 1967
14. Schagen, P., *X-ray image intensifiers: design and future possibilities*, in Ultrasound and X-rays in Engineering and Medicine, Royal Society, London, 1979
15. Takahashi, Y., et al., *Apparatus for detecting foreign matter with high selectivity and high sensitivity by image processing*, US Patent 6023497, 2000
16. Ter-pogossian, M., *The physical aspects of diagnostic radiology*, Harper and Row, USA, 1969
17. Webb, S. et al., *Physics of medical imaging*, Adam Hilger, 1988

APPENDIX B

IMPLEMENTATION DETAILS OF IMAGE PROCESSING AND CODE

B.1 Implementation details

As it was mentioned before in the main body of the thesis, all implementations were done into Borland Delphi 3.0 programming environment. Most of the implementation uses the same conventions and notations as in chapter 4 for all of the segmentation algorithms. The main important procedures are:

- `Histogram()` – procedure that computes the histogram values for both low- and high-energy images;
- `Init_Network()` and `Init_Network2()` – procedure for the HNN initialisation for segmentation of single band and dual band images respectively;
- `Hopfield()` and `Hopfield2()` – the main procedures for the minimization of the associated energy function for segmentation of a single-band and of a dual-band image;
- `Vignetting()` – procedure that simulates the vignetting effect on both high and low-energy images;
- `Read_Input()` –procedure for reading the input images; moreover, the procedure can add impulse noise to the input images if required;
- `WTA()` – function that implement the Winner Takes All algorithm;
- `Hopfield_Koss()` – procedure for the implementation of HNN based segmentation algorithm after *[Koss et al., 1999]* incorporating spatial constraints in the definition of the energy associated with the NN; it uses procedures like `Calculate_R()` that computes the centroids of a point to a class, and `Init_Network_Koss()` that initialises the HNN in this case;

- FCM() – the implementation of the Fuzzy C-Means Clustering algorithm

The main procedure of the HNN segmentation algorithm is presented below.

Implementation of HNN in Borland Delphi

```
procedure Hopfield2;
var i,j,x,y:integer;
    sum1,sum2:double;
begin
Histogram; //compute the histograms for bothe low and high energy images
if (no_greys<no_greys2) then no_greys:=no_greys2; //determine the
maximum number of level of grey present in the dual image

Init_network2; //randomly initialize the network
repeat //repeat until there are no changing neurons
//save the old values for the neurons output
for i:= to no_greys do
    for j:= to k do V_old[i,j]:=V[i,j];
for x:= to no_greys do
    begin
          //calculate the input for row x
          sum1:= ;
          sum2:= ;
          for i:= to k do
            begin
              for y:= to no_greys do
                        sum1:=sum1+(h[y]+h2[y])*V[y,i];
              for y:= to no_greys do
                        sum2:=sum2+DIS[x,y]*(h[y]+h2[y])*V[y,i];
              Net[x,i]:=-( /sum1)*sum2;
            end; //end of input computations
           WTA(x); // apply the Winner Take All algorithm
    end;
// count the number of changed neurons
Count_changed_nodes;

until (count= ); //end the algorithm if the number of changed neurons
                                    is zero
        // there are no changing neurons

end; //end of Hopfield
```

B.2 Code

NOTE: All the code listed below represents the main part of the logical computation involved for HNN segmentation process, noise generation, FCM clustering algorithm and classical edge detection algorithms. Due to space constraints, all code related to the actual user interface and file input/output were left aside. Comments were provided

along the lines of code, were the author thought they were required. The software application can be found on the attached compact disk.

```
Type  convolution=array[0..2,0..2] of integer;
//the convolution type mask used for the classical edge detection algorithms

const
//constant convolution masks for Sobel, Prewitt, Gradient, etc. edge detection
      sobelh:convolution =((1,2,1),(0,0,0),(-1,-2,-1));
      sobelv:convolution =((1,0,-1),(2,0,-2),(1,0,-1));
      gradient:convolution =((-1,-1,-1),(1,-2,1),(1,1,1));
      laplace:convolution =((0,-1,0),(-1,4,-1),(0,-1,0));
      prewith:convolution =((-1,-1,-1),(0,0,0),(1,1,1));
      prewitv:convolution =((1,0,-1),(1,0,-1),(1,0,-1));

var
   Form1: TForm1;
   output, input,input2:array [0..536,0..536] of integer;
// input, output matrices
   net, net_old : array[0..287296, 0..20] of double;
// the Hopfield NN (only for 256 levels of grey)
   weights:array [1..10,1..256, 1..256] of  double;
//the weights of the NN
   h,h2,histo:array[0..255] of integer; //the histogram values
   no_greys,no_greys2:integer;
// the number of grey-levels present in the image
   k:integer; // the number of classes
   DIS:array[0..255,0..255] of integer; // array of square distance measures
   input_file, output_file, input2_file:text;
   segment_file:array[0..20] of text;
   input_file_name, output_file_name, input2_file_name:string;
   segment_file_name:array[0..20] of string;
   n:integer; //number of lines and columns of the specified X-ray text image
   min,max : integer;
//the maximum and minimum values of the grey level within the image
   V,V_old:array[0..287296,0..10] of integer;
// the output-input of WTA matrice
   count:integer; //the number of changed nodes
   iter:integer;// the number of algorithm's iterations
   grey_levels:array[0..255] of integer;
//the vector of grey_levels present in the image
   R:array[0..287296,0..20] of double;
   Xc,Yc:array[0..20] of double;
   noise_level,pixel_level:integer;
   noise:boolean;
   energy:double;
   en:array[0..50] of double;
//vector to save the nergy values for each iterations
   count_changed:array[0..50] of integer;
//vector to save the number of changed nodes for each iteration


//*****************************************************************************
//*****************************************************************************
//*****************************************************************************
procedure Vignetting;
// procedure for simulating the vignetting effect
var r,x,y,xc,yc,code:integer;
    new_grey:integer;
begin

xc:=n div 2;
yc:=n div 2;
```

```
new_grey:=0;

for r:=1 to (n div 2) do
begin
    for x:=0 to n do
    for y:=0 to n do
    begin
        if (sqr(x-xc)+sqr(y-yc)<=sqr(r))  then //is it within the circle with radius r?
           begin
               if  (r  mod  15=0)  and  (sqr(x-xc)+sqr(y-yc)>sqr(r-15))  then
new_grey:=r div 15
                  else new_grey:=0;                    //compute the new grey level
               input[x,y]:=input[x,y]+new_grey;
               input2[x,y]:=input2[x,y]+new_grey;
               if (input[x,y]>255) then input[x,y]:=255;
               if (input2[x,y]>255) then input2[x,y]:=255;
           end;
    end;
end;
end;



//*************************************************************************
//*************************************************************************
//*************************************************************************
procedure Init_network; //initialise the input to the network
var i,j:integer;
    max:double;
begin
    for i:=0 to no_greys do
    for j:=0 to k do
        net[i,j]:=random;

  //compute the maximum of each row and the initial V
    for i:=0 to no_greys do
    begin
         max:=Net[i,0];
         for j:=0 to k do
             if (Net[i,j]>max) then max:=Net[i,j];
         for j:=0 to k do
             if (Net[i,j]=max) then V[i,j]:=1
                               else V[i,j]:=0;
    end;
//update the V_old matrix
    for i:=0 to no_greys do
    for j:=0 to k do V_old[i,j]:=V[i,j];
end;



//*************************************************************************
//*************************************************************************
//*************************************************************************
procedure Init_network2;  //initialise the input to the network
var i,j,l:integer;
    maxim:double;
    alpha:double;
    c,temp:double;
    good:boolean;
    position,row:integer;
begin
// min and max are the minimum and maximum grey level values
// initialise network like in Poli and Valli 1997
alpha:=random;
    for i:=0 to no_greys do
```

```
    for j:=0 to k do
        begin
            c:=int((i*1.0-min*1.0)/(max*1.0-min*1.0)+1);
            net[i,j]:=0;
            if ((j=c-1) or (j=c+1)) then net[i,j]:=alpha;
            if (j=c) then net[i,j]:=1-2*alpha;
        end;


  //compute the maximum of each row and the initial V
    for i:=0 to no_greys do
    begin
        maxim:=Net[i,0];
        for j:=0 to k do
            if (Net[i,j]>maxim) then maxim:=Net[i,j];
        for j:=0 to k do
            if (Net[i,j]=maxim) then V[i,j]:=1
                                else V[i,j]:=0;
    end;

//make sure that every class contains at least one grey level
//on every column there should be at least one 1 (in V) and a winner
repeat
for i:=0 to k do
begin
good:=false;
for j:=0 to no_greys do
        if (V[j,i]=1) then good:=true;
if (good=false) then //class i has no grey level
   begin
   //we need to change the winner of a random row
   //first we find out the winner
       row:=random(no_greys)+1;
       for l:=0 to k do if (V[row,l]=1) then position:=l;
   //we have old winner at (row,l)
   //change it with (row,i)
       temp:=net[row,position];
       net[row,position]:=net[row,i];
       net[row,i]:=temp;
   //change the values for the winner matrice as well
       V[row,i]:=1; //set the winner
       V[row,position]:=0; //loose the old winner
   end;
end;
until (good=true);

  //update the V_old matrix
    for i:=0 to no_greys do
    for j:=0 to k do V_old[i,j]:=V[i,j];
end;



//*****************************************************************************
//*****************************************************************************
//*****************************************************************************
procedure Count_changed_nodes;
var i,j:integer;
begin
    count:=0;
    for i:=0 to no_greys do
    for j:=0 to k do
        if (V[i,j]<>V_old[i,j]) then count:=count+1;
end;
```

```
//*************************************************************************
//*************************************************************************
//*************************************************************************
procedure WTA(x:integer);       // what if there are 2 nodes with the same value
var i:integer;
    max:double;
begin
    max:=Net[x,0];
    for i:=1 to k do
        if (Net[x,i]>max) then max:=Net[x,i];
    for i:=0 to k do
        if (Net[x,i]=max) then V[x,i]:=1
                          else V[x,i]:=0;
end;


//*************************************************************************
//*************************************************************************
//*************************************************************************
procedure Histogram; // compute the histogram values
var i,j:integer;
begin
for i:=0 to 255 do
    begin
    h[i]:=0;
    h2[i]:=0;
    end;

for i:=0 to n do
for j:=0 to n do
         begin
         h[input[i,j]]:=h[input[i,j]]+1; //compute the histogram values
         h2[input2[i,j]]:=h[input2[i,j]]+1;
         end;
// compute the number of grey-levels present in the image
no_greys:=0;
no_greys2:=0;
j:=0;
for i:=0 to 255 do
    if (h[i]<>0) then
                    begin
                    no_greys:=no_greys+1;
                    grey_levels[j]:=i;
                    histo[j]:=h[i];
                    j:=j+1;
                    end;
for i:=0 to 255 do h[i]:=histo[i];

j:=0;
for i:=0 to 255 do
    if (h2[i]<>0) then
                    begin
                    no_greys2:=no_greys2+1;
                    histo[j]:=h[i];
                    j:=j+1;
                    end;
for i:=0 to 255 do h2[i]:=histo[i];


//compute the set of all the square distance measures (gx-gy)2
for i:=0 to 255 do
for j:=0 to 255 do
    begin
        DIS[i,j]:=sqr(i-j);
    end;
end;
```

```
//compute the optimum number of classes for that particular image
procedure OptimalNumberOfClasses;
var i,j,l:integer;
    hi,Hf:array[0..255]of real;
    Hj:array[0..20,0..255]of real;
    mu,f,sigma,err:array[0..20]of real;
    sum1,tmp1,min:real;
    optim:integer;

begin
sum1:=0;
for i:=0 to no_greys do
    sum1:=sum1+h[i];
for i:=0 to no_greys do
    hi[i]:=h[i]/sum1;
//now we have maximum 10 classes, therefore we divide the number of greys present
//in the image with 10 and we have now ten clusters

for j:=1 to 10 do
    begin
    f[j]:=0;
    mu[j]:=0;
    sigma[j]:=0;
      for i:=0 to no_greys do
        if (i>=(j-1)*(no_greys/10)) and (i<j*(no_greys/10)) then
                                    f[j]:=f[j]+hi[i];
      for i:=0 to no_greys do
        if (i>=(j-1)*(no_greys/10)) and (i<j*(no_greys/10)) then
                                    mu[j]:=mu[j]+i*hi[i];
        mu[j]:=mu[j]/f[j];
      for i:=0 to no_greys do
        if (i>=(j-1)*(no_greys/10)) and (i<j*(no_greys/10)) then
                                    sigma[j]:=sigma[j]+(sqr(i-mu[j])*hi[i]);
        sigma[j]:=sigma[j]/f[j];
    end;

for l:=0 to no_greys do
for j:=1 to 10 do
    Hj[j][l]:=1/(sigma[j]*sqrt(2*PI))*exp(-sqr(l-mu[j])/(2*sqr(sigma[j])));
for l:=0 to no_greys do
    begin
    Hf[l]:=0;
    for j:=1 to 10 do
        Hf[l]:=Hf[l]+f[j]*Hj[j][l];
    end;

for j:=1 to 10 do
    begin
        err[j]:=0;
       for l:=0 to no_greys do
           err[j]:=err[j]+sqr(hi[l]-Hf[l]);
    end;
//find the minimum now
min:=err[1];
for j:=1 to 10 do
     if err[j]<=min then begin min:=err[j];i:=j;end;
optim:=i;
end;



//*************************************************************************
//*************************************************************************
```

```
//*************************************************************************
procedure Hopfield;
var i,j,x,y:integer;
    sum1,sum2:double;
    sir:string;
begin
iter:=0;
Histogram;
Init_network;
repeat //repeat until there are no changing neurons
//save the old values for the neurons output
for i:=0 to no_greys do
    for j:=0 to k do V_old[i,j]:=V[i,j];
for x:=0 to no_greys do
    begin
    //calculate the input for row x
          sum1:=0;
          sum2:=0;
          for i:=0 to k do
            begin
              for y:=0 to no_greys do
                       sum1:=sum1+h[y]*V[y,i];
              for y:=0 to no_greys do
                       sum2:=sum2+DIS[x,y]*h[y]*V[y,i];
              Net[x,i]:=-(1/sum1)*sum2;
            end; //end of input computations
           WTA(x);
    end;
// count the number of changed neurons
Count_changed_nodes;
iter:=iter+1;
until (count=0); //end the algorithm if the number of changed neurons is zero
        // there are no changing neurons
end;



//*************************************************************************
//*************************************************************************
//*************************************************************************
procedure Hopfield2;
var i,j,x,y:integer;
    sum1,sum2:double;
    sir:string;
begin
iter:=0;
Histogram;

OptimalNumberOfClasses;

if (no_greys<no_greys2) then no_greys:=no_greys2;

Init_network2;
repeat //repeat until there are no changing neurons
//save the old values for the neurons output
for i:=0 to no_greys do
    for j:=0 to k do V_old[i,j]:=V[i,j];
for x:=0 to no_greys do
    begin
    //calculate the input for row x
          sum1:=0;
          sum2:=0;
          for i:=0 to k do
            begin
              for y:=0 to no_greys do
                       sum1:=sum1+(h[y]+h2[y])*V[y,i];
```

```
                for y:=0 to no_greys do
                        sum2:=sum2+DIS[x,y]*(h[y]+h2[y])*V[y,i];
                Net[x,i]:=-(1/sum1)*sum2;
              end; //end of input computations
            WTA(x);
      end;
 // count the number of changed neurons
 Count_changed_nodes;

 iter:=iter+1;
 str(iter,sir);
 until (count=0); //end the algorithm if the number of changed neuronsis zero
         // there are no changing neurons
 end; //end of Hopfield 2



 //*************************************************************************
 //*************************************************************************
 //*************************************************************************
 procedure Read_Input;
 var i,j,x,y,code:integer;
     sir:string;
     temp,temp1:double;
     noise_mask:array[0..536,0..536] of boolean;
 begin
 // the number of pixels in the image (per column or row)
 min:=0;
 max:=255;
 for i:=0 to n do
 for j:=0 to n do
       begin
            read(input_file, input[i,j]);
            read(input2_file, input2[i,j]);
            if (input[i,j]<min) then min:=input[i,j];
            if (input[i,j]>max) then max:=input[i,j];
       end;
//generate noise
if noise then
begin
     randomize;
     for  i:=0 to n do
     for j:=0 to n do noise_mask[i,j]:=false;

     for i:=0 to ((n*n*pixel_level) div 100) do
              begin
              repeat
       //generate random coordinates
                        x:=random(n+1);
                        y:=random(n+1);
              until not noise_mask[x,y];
                    noise_mask[x,y]:=true;
                       temp:=1-random(3);
                       temp1:=temp*input[x,y]*noise_level/100;
                       input[x,y]:=round(input[x,y]+temp1);
                       temp1:=temp*input2[x,y]*noise_level/100;
                       input2[x,y]:=round(input2[x,y]+temp1);
                       if (input[x,y]>255) then input[x,y]:=255;
                       if (input2[x,y]>255) then input2[x,y]:=255;
                       if (input[x,y]<0) then input[x,y]:=-input[x,y];
                       if (input2[x,y]<0) then input2[x,y]:=-input2[x,y];
              end;
close(input_file);
close(input2_file);
end;
```

```
//*************************************************************************
//*************************************************************************
//*************************************************************************
//initialise the input to the network for KOSSSSSSS
procedure Init_network_Koss;
var i,j:integer;
    max:double;
begin
    for i:=0 to (n+1)*(n+1)-1 do
    for j:=0 to k do
        net[i,j]:=random;

  //compute the maximum of each row and the initial V
    for i:=0 to (n+1)*(n+1)-1 do
    begin
         max:=Net[i,0];
         for j:=0 to k do
             if (Net[i,j]>max) then max:=Net[i,j];
         for j:=0 to k do
             if (Net[i,j]=max) then V[i,j]:=1
                               else V[i,j]:=0;
    end;
//update the V_old matrix
    for i:=0 to (n+1)*(n+1)-1 do
    for j:=0 to k do V_old[i,j]:=V[i,j];
end;



//*************************************************************************
//*************************************************************************
//*************************************************************************
procedure Calculate_R;
var p,i,j,xp,yp,coord_x,coord_y:integer;
    sum1,sum2,sum3:extended;
begin
//calculate the centroid (xp,yp)
for p:=0 to k do
    begin
        sum1:=0;
        sum2:=0;
        sum3:=0;
        for j:=0 to (n+1)*(n+1)-1 do
   // find out the coordinate of neuron Net[i,p];
            begin
            coord_x:=j div (n+1);
            coord_y:=j mod (n+1);
            sum1:=sum1+coord_x*V[j,p]; // or V[j,p]
            sum2:=sum2+V[j,p]; // or V[j,p]
            sum3:=sum3+coord_y*V[j,p]; // or V[j,p]
            end;
            Xc[p]:=sum1/sum2;
            Yc[p]:=sum3/sum2;
    end;
// calculate R[p,i] the distance of point i from centroid of class p
for i:=0 to (n+1)*(n+1)-1 do
for p:=0 to k do
    begin
    coord_x:=i div (n+1);
    coord_y:=i mod (n+1);
    R[i,p]:=sqr(coord_x-Xc[p])+sqr(coord_y-Yc[p]);
    end;
end;
```

```
//*************************************************************************
//*************************************************************************
//*************************************************************************
//Hopfield algorithm -KOSS
procedure Hopfield_Koss;
var i,j,p,x,y:integer;
    sum1,sum2:double;
    sir:string;
    max:double;
begin
iter:=0;
Init_network_Koss;
repeat //repeat until there are no changing neurons
//save the old values for the neurons output
for i:=0 to (n+1)*(n+1)-1 do
    for j:=0 to k do begin
                        V_old[i,j]:=V[i,j];
                        Net_old[i,j]:=Net[i,j];
                        end;

// first we calculate Rpi
Calculate_R;
// calculate the new value sof neurons using Net=RpiVpi
for i:=0 to (n+1)*(n+1)-1 do
for p:=0 to k do
    Net[p,i]:=R[p,i]*Net_old[p,i]; //or is it Net[i,p] ???
// count the number of changed neurons
// apply WTA scheme
    for i:=0 to (n+1)*(n+1)-1 do
    begin
        max:=Net[i,0];
        for j:=0 to k do
            if (Net[i,j]>max) then max:=Net[i,j];
        for j:=0 to k do
            if (Net[i,j]=max) then V[i,j]:=1
                              else V[i,j]:=0;
    end;

Count_changed_nodes;
iter:=iter+1;
until (count=0); //end the algorithm if the number of changed neuronsis zero
        // there are no changing neurons
end; //end of Hopfield_KOSS


//start of fuzzy algoritms FCM


procedure FCM;
var i,j,x,y:integer;
    stop:boolean;
    sir:string;
    U,U_old,difference:array[0..255,0..20] of real; // matrices for segmentation
    z:array[0..50] of real;      // vector of center of classes
    distance:array[0..255,0..20] of real; //the matrix for the distance measure
    m:byte; //the exponential weight (fuzzification parameter)
    temp1,temp2:real;
    max:real;
    err:real; //threshold error for convergence
begin

err:=0.2;

//STEP 1
//initialize arbitrarily the class centers
```

```
for i:=0 to k do
    z[i]:=random(no_greys)+1;
```

//STEP 2 - compute the histogram values already one in the ButtonClicked proc.

```
stop:=false;
m:=2;
iter:=0;
repeat
```

//STEP 3
//compute the distance between the gray level gx and the clas center zi for all classes:

```
for x:=0 to no_greys do
for i:=0 to k do
    begin
         distance[x,i]:=sqr(x-z[i]);
    end;
```

//save the values into U_old;

```
for x:=0 to no_greys do
for i:=0 to k do
            U_old[x,i]:=U[x,i];
```

//STEP 4
//calculate the membership matrix U

```
for x:=0 to no_greys do
for i:=0 to k do
begin
```

//for gx<>zx

```
   if (x<>z[i]) then
   begin
        temp1:=1/distance[x,i];
        temp2:=0;
        for j:=0 to k do
                 begin
                  if (x<>z[j]) then temp2:=temp2+1/distance[x,j];
                 end;
        U[x,i]:=temp1/temp2;
   end
   else
    if (x=i) then U[x,i]:=1
    else U[x,i]:=0;



end;
```

//STEP 5
//update class centers as follows

```
temp1:=0;
temp2:=0;
for i:=0 to k do
    begin
         temp1:=0;
         temp2:=0;
        for x:=0 to no_greys do
           begin
            temp1:=temp1+sqr(U[x,i])*h[x];
            temp2:=temp2+sqr(U[x,i])*x*h[x];
           end;
    z[i]:=int((1/temp1)*temp2);
    end;
```

//STEP 6
//compute the differences between matrices and check convergence

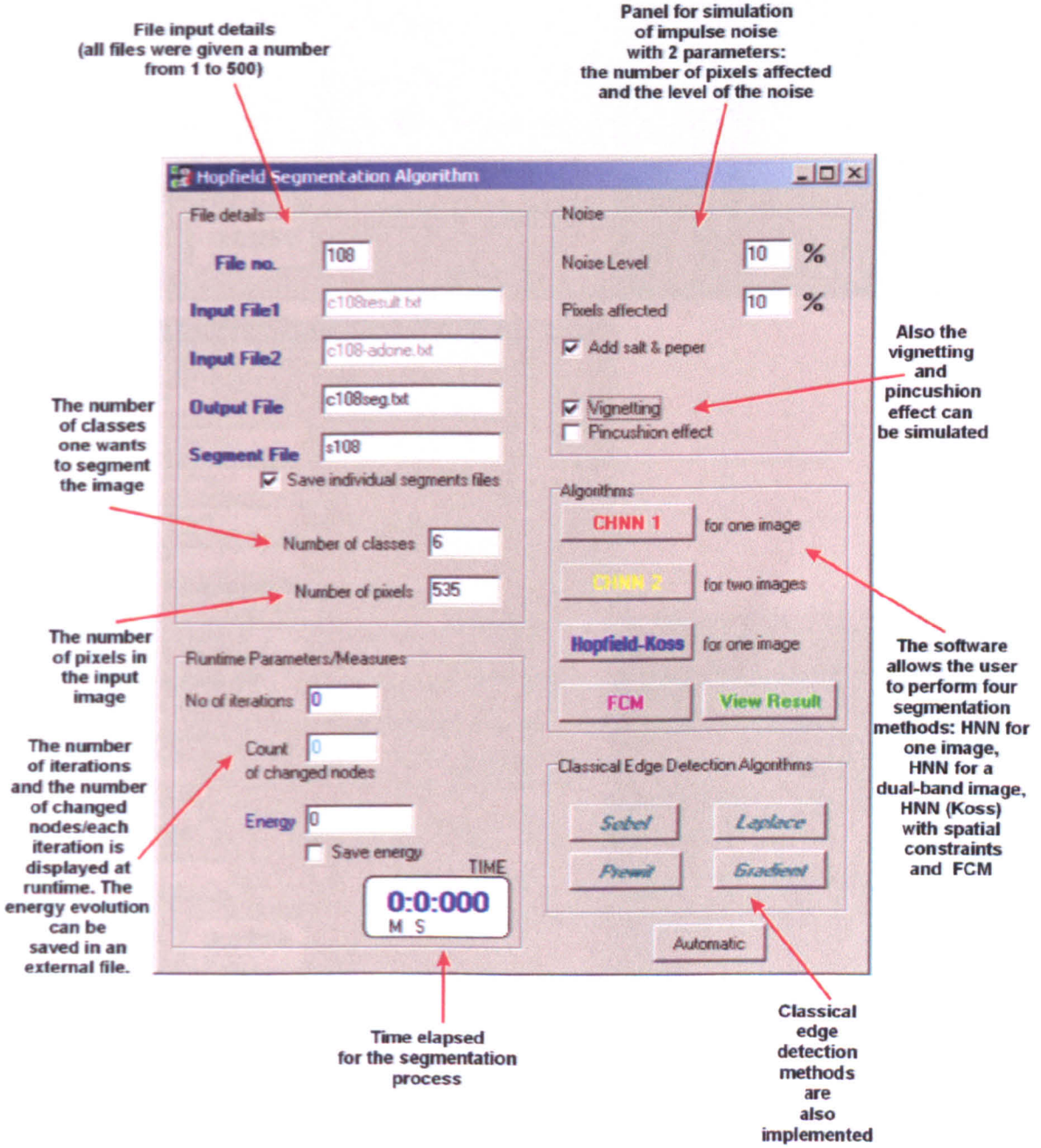
```
max:=-0.000001;
for x:=0 to no_greys do
for i:=0 to k do
begin
     difference[x,i]:=U[x,i]-U_old[x,i];
     if (abs(difference[x,i])>max) then max:=difference[x,i]; //compute maximum
end; //of STEP 6
if (max<err) then stop:=true;
//end of STEP6
iter:=iter+1;
str(iter,sir);
until stop;


//apply WTA
for x:=0 to no_greys do
begin
    max:=U[x,0];
    for i:=1 to k do
        if (U[x,i]>max) then max:=U[x,i];
    for i:=0 to k do
        if (U[x,i]=max) then V[x,i]:=1
                        else V[x,i]:=0;
end;

end; //end of FCM procedure



//*************************************************************************
// SOBEL edge detection algorithm
//*************************************************************************
procedure Sobel;
var i,j:integer;
begin
 for i:=2 to n-2 do
 for j:=2 to n-2 do
     begin
         input2[i,j]:=input[i-1,j-1]*sobelh[0,0]+
                      input[i,j-1]*sobelh[1,0]+
                      input[i+1,j-1]*sobelh[2,0]+
                      input[i-1,j]*sobelh[0,1]+
                      input[i,j]*sobelh[1,1]+
                      input[i+1,j]*sobelh[2,1]+
                      input[i-1,j+1]*sobelh[0,2]+
                      input[i,j+1]*sobelh[1,2]+
                      input[i+1,j+1]*sobelh[2,2];

         output[i,j]:=input[i-1,j-1]*sobelv[0,0]+
                      input[i,j-1]*sobelv[1,0]+
                      input[i+1,j-1]*sobelv[2,0]+
                      input[i-1,j]*sobelv[0,1]+
                      input[i,j]*sobelv[1,1]+
                      input[i+1,j]*sobelv[2,1]+
                      input[i-1,j+1]*sobelv[0,2]+
                      input[i,j+1]*sobelv[1,2]+
                      input[i+1,j+1]*sobelv[2,2];
     end;

 for i:=2 to n-2 do
 for j:=2 to n-2 do
     begin
          output[i,j]:=round(sqrt(sqr(input2[i,j])+sqr(input2[i,j])));
     end;
end;// end of Sobel procedure
```

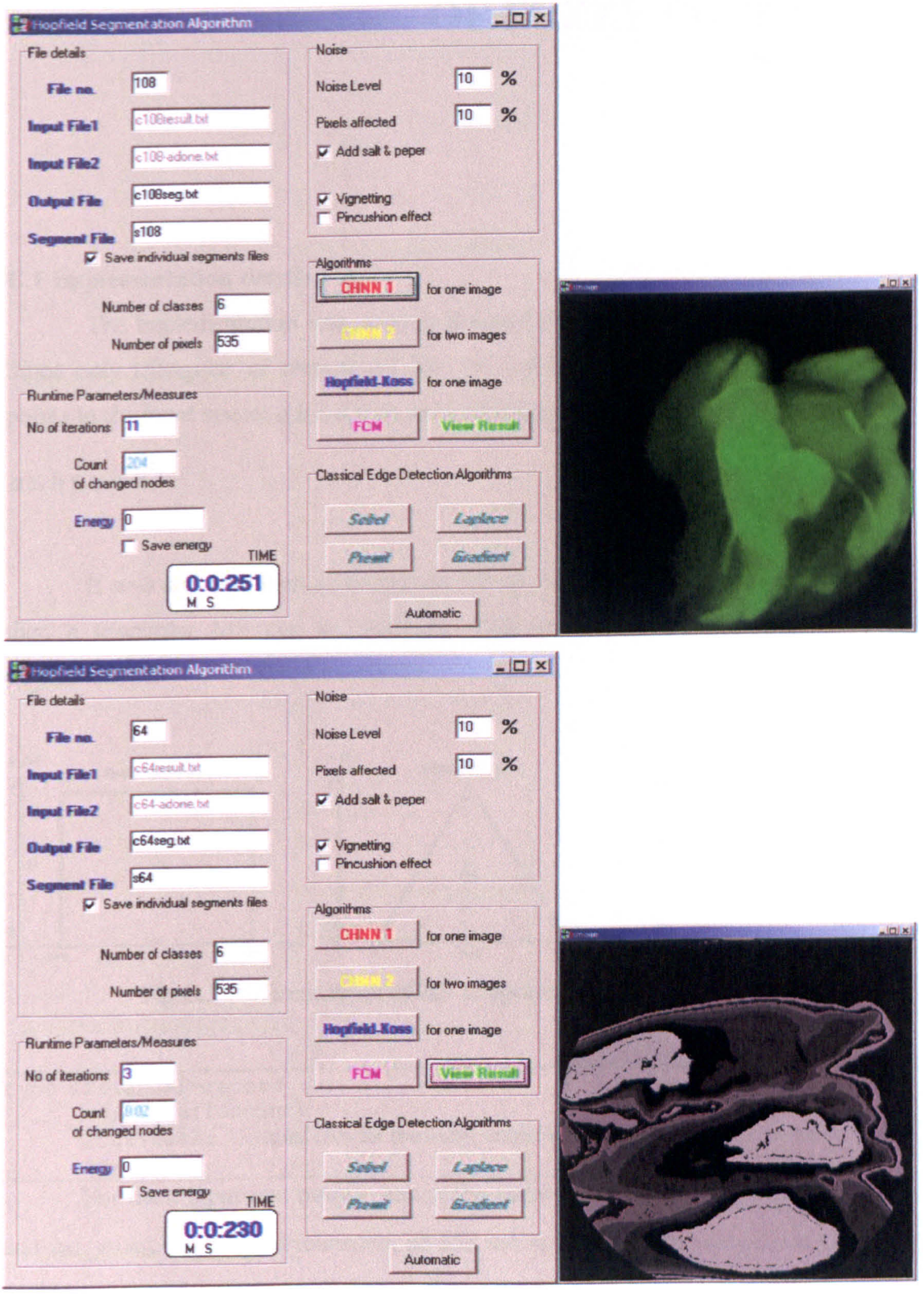
```
//*************************************************************************
// PREWIT edge detection algorithm
//*************************************************************************
procedure Prewit;
var i,j:integer;
begin
 for i:=2 to n-2 do
 for j:=2 to n-2 do
     begin
         input2[i,j]:=input[i-1,j-1]*prewith[0,0]+
                      input[i,j-1]*prewith[1,0]+
                      input[i+1,j-1]*prewith[2,0]+
                      input[i-1,j]*prewith[0,1]+
                      input[i,j]*prewith[1,1]+
                      input[i+1,j]*prewith[2,1]+
                      input[i-1,j+1]*prewith[0,2]+
                      input[i,j+1]*prewith[1,2]+
                      input[i+1,j+1]*prewith[2,2];

         output[i,j]:=input[i-1,j-1]*prewitv[0,0]+
                      input[i,j-1]*prewitv[1,0]+
                      input[i+1,j-1]*prewitv[2,0]+
                      input[i-1,j]*prewitv[0,1]+
                      input[i,j]*prewitv[1,1]+
                      input[i+1,j]*prewitv[2,1]+
                      input[i-1,j+1]*prewitv[0,2]+
                      input[i,j+1]*prewitv[1,2]+
                      input[i+1,j+1]*prewitv[2,2];
     end;

 for i:=2 to n-2 do
 for j:=2 to n-2 do
     begin
          output[i,j]:=round(sqrt(sqr(input2[i,j])+sqr(input2[i,j])));
     end;
end;// end of Prewit procedure


//*************************************************************************
// LAPLACE edge detection algorithm
//*************************************************************************
procedure Laplace1;
var i,j:integer;
begin
 for i:=2 to n-2 do
 for j:=2 to n-2 do
     begin
         output[i,j]:=input[i-1,j-1]*laplace[0,0]+
                      input[i,j-1]*laplace[1,0]+
                      input[i+1,j-1]*laplace[2,0]+
                      input[i-1,j]*laplace[0,1]+
                      input[i,j]*laplace[1,1]+
                      input[i+1,j]*laplace[2,1]+
                      input[i-1,j+1]*laplace[0,2]+
                      input[i,j+1]*laplace[1,2]+
                      input[i+1,j+1]*laplace[2,2];
     end;
end;// end of Laplace1 procedure


//*************************************************************************
// GRADIENT edge detection algorithm
//*************************************************************************
procedure Gradient1;
var i,j:integer;
begin
 for i:=2 to n-2 do
```

```
for j:=2 to n-2 do
    begin
        output[i,j]:=input[i-1,j-1]*gradient[0,0]+
                     input[i,j-1]*gradient[1,0]+
                     input[i+1,j-1]*gradient[2,0]+
                     input[i-1,j]*gradient[0,1]+
                     input[i,j]*gradient[1,1]+
                     input[i+1,j]*gradient[2,1]+
                     input[i-1,j+1]*gradient[0,2]+
                     input[i,j+1]*gradient[1,2]+
                     input[i+1,j+1]*gradient[2,2];
    end;
```

end;// end of Gradient1 procedure

B.3 Screenshots




Hopfield Segmentation Algorithm
File details
File no. 108
Input File1 c108result.txt
Input File2 c108-adone.txt
Output File c108seg.txt
Segment File s108
Save individual segments files
Number of classes 6
Number of pixels 535
Runtime Parameters/Measures
No of iterations 11
Count of changed nodes 204
Energy 0
Save energy
TIME
0:0:251
M S
Noise
Noise Level 10 %
Pixels affected 10 %
Add salt & peper
Vignetting
Pincushion effect
Algorithms
CHNN 1 for one image
CHNN 2 for two images
Hopfield-Koss for one image
FCM View Result
Classical Edge Detection Algorithms
Sobel Laplace
Prewit Gradient
Automatic
Hopfield Segmentation Algorithm
File details
File no. 64
Input File1 c64result.txt
Input File2 c64-adone.txt
Output File c64seg.txt
Segment File s64
Save individual segments files
Number of classes 6
Number of pixels 535
Runtime Parameters/Measures
No of iterations 3
Count of changed nodes 0.02
Energy 0
Save energy
TIME
0:0:230
M S
Noise
Noise Level 10 %
Pixels affected 10 %
Add salt & peper
Vignetting
Pincushion effect
Algorithms
CHNN 1 for one image
CHNN 2 for two images
Hopfield-Koss for one image
FCM View Result
Classical Edge Detection Algorithms
Sobel Laplace
Prewit Gradient
Automatic

APPENDIX C

IMPLEMENTATION DETAILS OF FUZZY PROCESSING AND CODE

C.1 Implementation details

The implementation was done in Borland Delphi 3.0 programming environment. Since only triangular or trapezoidal MF are used, the MF type can be defined by 4 points in the input space: a,b,c,d with the following convention:

$$a \le b \le c \le d \tag{c.1}$$

If $a=b$ and $c=d$ then a rectangular MF is obtained as shown in figure C.1. If $b=c$ then a triangular MF can be achieved. Different values of a,b,c and d render a trapezoidal MF.

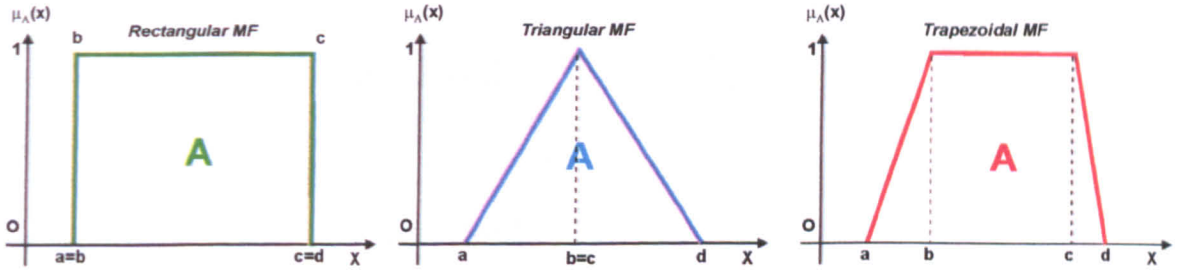


Fig.C.1 Different types of MF supported by the implementation

```
type TTriangMF=record
       a,b,c,d:longint;
       cut:real;     //used only for the fuzzy_output sets for each rule
     end;
```

For each input and output vector the number of MF is known `no_input_mf` and `no_output_mf`. Furthermore, all MF are specified in an array of vector for both inputs and outputs: `input_mf` and `output_mf`.

```
crisp_input:array[0..20] of real;
```

```
fuzzy_input:array[0..20,0..10] of real; //vectors for input fuzzy and crisp values
crisp_output:real;
no_input_mf:array[0..20]of integer; //the no. of mf for each input
no_output_mf:integer;
input_mf:array[0..20,0..4]of TTriangMF; //mf functions for inputs
output_mf:array[0..20] of TTriangMF;    //mf for output
fuzzy_output:array[0..20] of TTriangMF;
output_vector:array[0..10000]of real; //to store all the values of
                                      the resulting aggregated fuzzy output set
no_rules_ant:array[0..20]of integer;
rule:array[0..20] of TRule;
```

An example of how the HOLE measure was defined is illustrated below (fig. C.2). There are two linguistic variables defined for HOLE: SMALL and BIG:

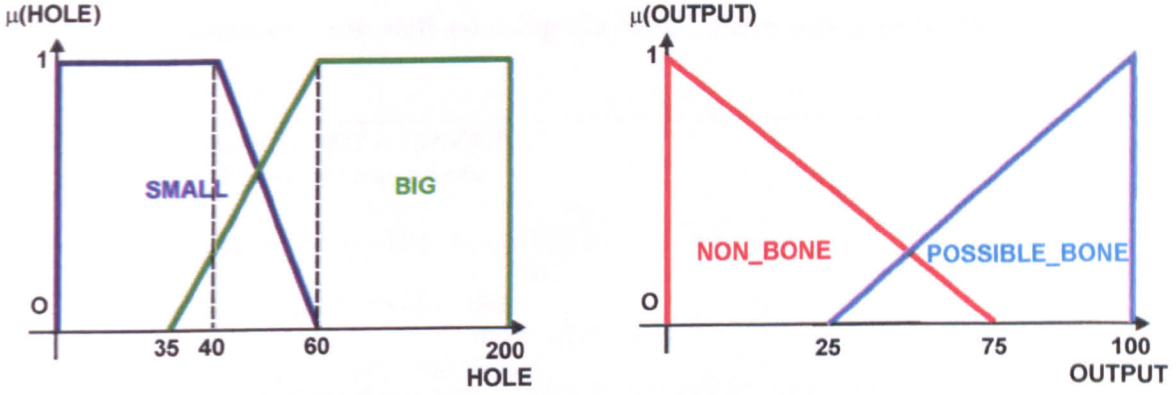


Fig. C.2 Fuzzy sets for HOLE and OUTPUT measures

```
input_mf[1,0].a:=0; //HOLE SMALL
input_mf[1,0].b:=0;
input_mf[1,0].c:=40;
input_mf[1,0].d:=60;

input_mf[1,1].a:=35;   //HOLE BIG
input_mf[1,1].b:=60;
input_mf[1,1].c:=200;
input_mf[1,1].d:=200;
```

```
output_mf[0].a:=0;    //NO_BONE
output_mf[0].b:=0;
output_mf[0].c:=0;
output_mf[0].d:=75;

output_mf[1].a:=25;    //BONE
output_mf[1].b:=100;
output_mf[1].c:=100;
output_mf[1].d:=100;
```

The definition of rules is very important since it gives flexibility to the implementation. One can easily add or erase one or more rules simply by respecting some conventions.

```
type TRule=record
        ant_input_index,ant_mf_index:array[0..10] of byte;
        con_mf_index:byte; //the mf index
     end;
```

A rule of the form *x* is *Y* contains an array of antecedents indexes `ant_input_index`, array that contains the index of the input variable x and another array of MF indexes `ant_mf_index` that specifies the corresponding MF function of the input variable Y. The `con_mf_index` defines the desired output MF.

An example of a rule defined using the HOLE fuzzy sets defined previously is:

```
//if HOLE is BIG then OUTPUT is NOBONE
no_rules_ant[2]:=0; //1 antecedents
                                    //if
rule[2].ant_input_index[0]:=1; //HOLE
                                    //is
rule[2].ant_mf_index[0]:=1; //BIG
                                    //then OUTPUT is
rule[2].con_mf_index:=0; //NOBONE
```

C.2 Code

NOTE: All the code listed below represents the main part of the logical computation involved for object extraction, feature extraction and fuzzy processing. Due to space constraints, all code related to the actual user interface and file input/output were left aside. Comments were provided along the lines of code, were the author thought they were required. The software application can be found on the attached compact disk.

```
type TTriangMF=record
       a,b,c,d:longint;
       cut:real;              //used only for the fuzzy_output sets for each rule
     end;

     TRule=record
         ant_input_index,ant_mf_index:array[0..10]of byte;
         con_mf_index:byte; //the mf index
     end;

var
  Form1: TForm1;
//variables used for FUZZY REASONING
```

```
  no_inputs,no_rules:integer; //number of inputs and rules
  crisp_input:array[0..20] of real;
  fuzzy_input:array[0..20,0..10] of real; //vectors for input fuzzy and crisp values
  crisp_output:real;
  no_input_mf:array[0..20]of integer; //the number of mf for each input/output
  no_output_mf:integer;
  input_mf:array[0..20,0..4]of TTriangMF; //mf functions for inputs
  output_mf:array[0..20] of TTriangMF;    //mf for output
  fuzzy_output:array[0..20] of TTriangMF;
  output_vector:array[0..10000]of real;
//to store all the values of the resulting concatenated fuzzy output set
  no_rules_ant:array[0..20]of integer;
  rule:array[0..20] of TRule;
  rule_strength:array[0..20]of real; //the strengths of the rules

//variables used for CLASSICAL CONTOUR SORTING
  in1,in2,output, input,original, mask, final, copy, air:array [0..536,0..536]
of integer; // input, output matrices
  in1_file,in2_file,input_file, output_file, original_file:text;
  in1_file_name,in2_file_name,input_file_name,
      output_file_name,original_file_name:string;
  n:integer; //number of lines and columns of the specified X-ray text image
  contour_x:array [0..50,0..2000] of integer; //vectors for the contours (x,y) values
  contour_y:array [0..50,0..2000] of integer;
  k:integer; //the number of classes that the image was segmented into
  AIRHOLE_SIZE,MIN_BONE_SIZE,MAX_BONE_SIZE:integer;
                              //accepted sizes for the airholes and bones
  brightness1,brightness2,contrast1,contrast2,perimeter1,contrast,brightness:
                                                    real;
  roundness,togetherness1,togetherness2,togetherness,shape,
      difference1,difference2,difference:real;
  no_bones:integer;


//PROCEDURES AND FUNCTIONS FOR FUZZY COMPUTATIONS


function GetMFValue(index:integer;value:real;mf_index:integer):real;
begin

     if (value<input_mf[index,mf_index].a) then GetMFValue:=0;
     if (value>input_mf[index,mf_index].d) then GetMFValue:=0;
     if (value>=input_mf[index,mf_index].a)and
        (value<input_mf[index,mf_index].b) then
        begin
        if    (input_mf[index,mf_index].a=input_mf[index,mf_index].b)    then
GetMFValue:=1
        else
        GetMFValue:=(value-
input_mf[index,mf_index].a)/(input_mf[index,mf_index].b-
input_mf[index,mf_index].a);
        end;
     if (value>input_mf[index,mf_index].c)and
        (value<=input_mf[index,mf_index].d) then
        begin
        if    (input_mf[index,mf_index].c=input_mf[index,mf_index].d)    then
GetMFValue:=1
        else
        GetMFValue:=(input_mf[index,mf_index].d-
value)/(input_mf[index,mf_index].d-input_mf[index,mf_index].c);
        end;
     if (value>=input_mf[index,mf_index].b)and
        (value<=input_mf[index,mf_index].c) then
        GetMFValue:=1;
```

```
end;


procedure FuzzifyInput(index:integer);
var i:integer;
begin
     for i:=0 to no_input_mf[index] do
         fuzzy_input[index][i]:=GetMFValue(index,crisp_input[index],i);
end;



procedure EvaluateRule(rule_index:integer);
var i:integer;
    cutoff,a_val,b_val,c_val,d_val:real;
    x_val1,x_val2:real;
    val:array[0..10] of real;

begin

//evaluate the antecendents for the current rule
 for i:=0 to no_rules_ant[rule_index] do
     begin

val[i]:=fuzzy_input[rule[rule_index].ant_input_index[i]][rule[rule_index].ant
mf_index[i]];
     end;
//calculate the MINIMUM for the AND part of all those fuzzy antecedents
 cutoff:=val[0];
 for i:=0 to no_rules_ant[rule_index] do
         if (cutoff>val[i]) then cutoff:=val[i];
//calculate the minimum between CUTOFFF and the rule strength
 if rule_strength[rule_index]<=cutoff then cutoff:=rule_strength[rule_index];

//finished with the antecedents and we have the cutoff value now
//evaluate the consequents
 a_val:=output_mf[rule[rule_index].con_mf_index].a;
 b_val:=output_mf[rule[rule_index].con_mf_index].b;
 c_val:=output_mf[rule[rule_index].con_mf_index].c;
 d_val:=output_mf[rule[rule_index].con_mf_index].d;
 if (cutoff=0) then
                   begin
                        fuzzy_output[rule_index].a:=0;
                        fuzzy_output[rule_index].b:=0;
                        fuzzy_output[rule_index].c:=0;
                        fuzzy_output[rule_index].d:=0;
                        fuzzy_output[rule_index].cut:=cutoff;
                   end;
 if (cutoff=1) then // copy the exact output_mf[rule_index];
                   begin
                        fuzzy_output[rule_index].a:=round(a_val);
                        fuzzy_output[rule_index].b:=round(b_val);
                        fuzzy_output[rule_index].c:=round(c_val);
                        fuzzy_output[rule_index].d:=round(d_val);
                        fuzzy_output[rule_index].cut:=cutoff;
                   end;
 if (cutoff>0) and (cutoff<1) then
    begin
         if (a_val=b_val) then x_val1:=a_val
         else   x_val1:=b_val-(1-cutoff)*(b_val-a_val);
         if (c_val=d_val) then x_val2:=c_val
         else   x_val2:=c_val-(1-cutoff)*(c_val-d_val);

         fuzzy_output[rule_index].a:=round(a_val);
         fuzzy_output[rule_index].b:=round(x_val1);
         fuzzy_output[rule_index].c:=round(x_val2);
         fuzzy_output[rule_index].d:=round(d_val);
```

```
          fuzzy_output[rule_index].cut:=cutoff;

     end;

end;


function GetValue(value,a,b,c,d,cutoff:real):real;
begin
     if (value<a) then GetValue:=0;
     if (value>d) then GetValue:=0;
     if (value>a)and (value<b) then
        begin
        if (a=b) then GetValue:=cutoff
        else
        GetValue:=(value-a)/(b-a);
        end;
     if (value>c)and (value<=d) then
        begin
        if (c=d) then GetValue:=cutoff
        else
        GetValue:=(d-value)/(d-c);
        end;
     if (value>=b)and (value<=c) then
        GetValue:=cutoff;
end;


procedure DefuzzifyOutput;
var i,j,l:integer;
    val:array[0..10]of real;
    max:real;
    min_a,max_d:integer;
    total_area,sum:real;

begin
//concatenate the output fuzzy sets
//first find minimum a and maximum d
// then find the maximum values for the output vector of all output MF from the rules
min_a:=30000;
max_d:=0;
for i:=0 to no_rules do
    begin
         if (min_a>fuzzy_output[i].a) then min_a:=fuzzy_output[i].a;
         if (max_d<fuzzy_output[i].d) then max_d:=fuzzy_output[i].d;
    end;

//for each point between min_a and max_d compute the maximum value of the fuzzy sets
for i:=min_a to max_d do
    begin
         for j:=0 to no_rules do
             begin

val[j]:=GetValue(i,fuzzy_output[j].a,fuzzy_output[j].b,fuzzy_output[j].c,fuzzy
_output[j].d,fuzzy_output[j].cut);
             end;
         //get the maximum of all and store it in the output vector
         max:=0;
         for j:=0 to no_rules do
             begin
                if (max<val[j]) then max:=val[j];
             end;
         output_vector[i]:=max;
    end;
//now we have the concatenated output fuzzy set in the output vector from min_a to max_d
//we need to defuzzify the output
//centroid is in the following mode
```

```
total_area:=0;
sum:=0;
for i:=min_a to max_d do
    begin
    total_area:=total_area+output_vector[i];
    sum:=sum+i*output_vector[i];
    end;
if total_area<>0 then crisp_output:=sum/total_area
                      else crisp_output:=0;


PrintFuzzyResults(min_a,max_d);
Form1.Repaint;
end;



procedure Fuzzy_Initialise;
var sir:string;
    code:integer;
begin

no_inputs:=4;   //5 inputs AREA, HOLE, PERIMETER1, SHAPE, ROUNDNESS in this order
crisp_input[0]:=1102;       //AREA input value
crisp_input[1]:=45;       //HOLE input value
//the values on the top will be updated every time we make a FuzzyDecision with
//the measured values of the current area in question

no_input_mf[0]:=4; //5 mf for AREA
no_input_mf[1]:=1; //2 mf for HOLE
no_input_mf[2]:=1; //2 mf for PERIMETER
no_input_mf[3]:=1; //2 mf for SHAPE
no_input_mf[4]:=2; //3 mf for ROUNDNESS

//*************************************************************************
input_mf[0,0].a:=0;   //AREA TINY
input_mf[0,0].b:=0;
input_mf[0,0].c:=20;
input_mf[0,0].d:=30;

input_mf[0,1].a:=28;    //AREA SMALL
input_mf[0,1].b:=30;
input_mf[0,1].c:=165;
input_mf[0,1].d:=180;

input_mf[0,2].a:=170;    //AREA MEDIUM
input_mf[0,2].b:=180;
input_mf[0,2].c:=350;
input_mf[0,2].d:=400;

input_mf[0,3].a:=380;    //AREA BIG
input_mf[0,3].b:=500;
input_mf[0,3].c:=1300;
input_mf[0,3].d:=1500;

input_mf[0,4].a:=1400;   //AREA XLARGE
input_mf[0,4].b:=1500;
input_mf[0,4].c:=2000;
input_mf[0,4].d:=2000;

//*************************************************************************
input_mf[1,0].a:=0; //HOLE SMALL
input_mf[1,0].b:=0;
input_mf[1,0].c:=40;
input_mf[1,0].d:=60;
```

```
input_mf[1,1].a:=35;   //HOLE BIG
input_mf[1,1].b:=60;
input_mf[1,1].c:=200;
input_mf[1,1].d:=200;

//*************************************************************************
input_mf[2,0].a:=0; //PERIMETER SHORT
input_mf[2,0].b:=0;
input_mf[2,0].c:=320;
input_mf[2,0].d:=320;

input_mf[2,1].a:=280;   //PERIMETER LONG
input_mf[2,1].b:=280;
input_mf[2,1].c:=5000;
input_mf[2,1].d:=5000;

//*************************************************************************
input_mf[3,0].a:=0; //SHAPE IRREGULAR
input_mf[3,0].b:=0;
input_mf[3,0].c:=15;
input_mf[3,0].d:=15;

input_mf[3,1].a:=14;   //SHAPE SMOOTH
input_mf[3,1].b:=14;
input_mf[3,1].c:=100;
input_mf[3,1].d:=100;
//*************************************************************************

input_mf[4,0].a:=0; //ROUNDNESS LOW
input_mf[4,0].b:=0;
input_mf[4,0].c:=18;
input_mf[4,0].d:=20;

input_mf[4,1].a:=19; //ROUNDNESS MEDIUM
input_mf[4,1].b:=20;
input_mf[4,1].c:=65;
input_mf[4,1].d:=78; //65 /78

input_mf[4,2].a:=68;   //ROUNDNESS HIGH 68/78
input_mf[4,2].b:=78;
input_mf[4,2].c:=100;
input_mf[4,2].d:=100;
//*************************************************************************


//define the output mf
no_output_mf:=1; //2 mf for output

output_mf[0].a:=0;    //NO_BONE
output_mf[0].b:=0;
output_mf[0].c:=0;
output_mf[0].d:=75;

output_mf[1].a:=25;    //BONE
output_mf[1].b:=100;
output_mf[1].c:=100;
output_mf[1].d:=100;


//define the rules
no_rules:=10; //11 rules 0 and 1,2,3,4...9

//*****************************************************************
//if AREA is TINY then OUTPUT is NOBONE
no_rules_ant[0]:=0; //1 antecedents
                                        //if
```

```
rule[0].ant_input_index[0]:=0; //AREA
                                //is
rule[0].ant_mf_index[0]:=0; //TINY
                              //then OUTPUT is
rule[0].con_mf_index:=0; //NOBONE

rule_strength[0]:=1;
//*****************************************************************
//if AREA is XLARGE then OUTPUT is NO_BONE
no_rules_ant[1]:=0; //1 ant
                                //if
rule[1].ant_input_index[0]:=0;  //AREA
                                //is
rule[1].ant_mf_index[0]:=4;     //XLARGE
                                //then  OUTPUT is
rule[1].con_mf_index:=0;        //NO_BONE

rule_strength[1]:=1;
//*****************************************************************
//if HOLE is BIG then OUTPUT is NOBONE
no_rules_ant[2]:=0; //1 antecedents
                                  //if
rule[2].ant_input_index[0]:=1; //HOLE
                                //is
rule[2].ant_mf_index[0]:=1; //BIG
                              //then OUTPUT is
rule[2].con_mf_index:=0; //NOBONE

rule_strength[2]:=1;
//*****************************************************************
//if SHAPE is IRREGULAR then OUTPUT is NOBONE
no_rules_ant[3]:=0; //1 antecedents
                                //if
rule[3].ant_input_index[0]:=3; //SHAPE
                                //is
rule[3].ant_mf_index[0]:=0; //IRREGULAR
                              //then OUTPUT is
rule[3].con_mf_index:=0; //NOBONE

rule_strength[3]:=1;
//*****************************************************************
//if PERIMETER is LONG then OUTPUT is NOBONE
no_rules_ant[4]:=0; //1 antecedents
                                //if
rule[4].ant_input_index[0]:=2; //PERIMETER
                                //is
rule[4].ant_mf_index[0]:=1; //LONG
                              //then OUTPUT is
rule[4].con_mf_index:=0; //NOBONE

rule_strength[4]:=1;
//*****************************************************************
//if AREA is BIG and HOLE is SMALL and SHAPE is SMOOTH and
//      PERIMETER is SHORT and ROUNDNESS is HIGH then OUTPUT is NOBONE
no_rules_ant[5]:=4; //5 antecedents
                                //if
rule[5].ant_input_index[0]:=0; //AREA
                                //is
rule[5].ant_mf_index[0]:=3; //BIG
                              //and
rule[5].ant_input_index[1]:=1; //HOLE
                                //is
rule[5].ant_mf_index[1]:=0; //SMALL
                              //and
```

```
rule[5].ant_input_index[2]:=3; //SHAPE
                                //is
rule[5].ant_mf_index[2]:=1; //SMOOTH
                                //and
rule[5].ant_input_index[3]:=2; //PERIMETER
                                //is
rule[5].ant_mf_index[3]:=0; //SHORT
                                //and
rule[5].ant_input_index[4]:=4; //ROUNDNESS
                                //is
rule[5].ant_mf_index[4]:=2; //HIGH
                                //then OUTPUT is
rule[5].con_mf_index:=0; //NO_BONE

rule_strength[5]:=1;
//*****************************************************************
//if AREA is MEDIUM and HOLE is SMALL and SHAPE is SMOOTH and
//      PERIMETER is SHORT and ROUNDNESS is HIGH then OUTPUT is NOBONE
no_rules_ant[6]:=4; //5 antecedents
                                //if
rule[6].ant_input_index[0]:=0; //AREA
                                //is
rule[6].ant_mf_index[0]:=2; //MEDIUM
                                //and
rule[6].ant_input_index[1]:=1; //HOLE
                                //is
rule[6].ant_mf_index[1]:=0; //SMALL
                                //and
rule[6].ant_input_index[2]:=3; //SHAPE
                                //is
rule[6].ant_mf_index[2]:=1; //SMOOTH
                                //and
rule[6].ant_input_index[3]:=2; //PERIMETER
                                //is
rule[6].ant_mf_index[3]:=0; //SHORT
                                //and
rule[6].ant_input_index[4]:=4; //ROUNDNESS
                                //is
rule[6].ant_mf_index[4]:=2; //HIGH
                                //then OUTPUT is
rule[6].con_mf_index:=0; //NO_BONE

rule_strength[6]:=1;

//*****************************************************************
//if AREA is SMALL and HOLE is SMALL and SHAPE is SMOOTH and
//      PERIMETER is SHORT then OUTPUT is BONE
no_rules_ant[7]:=3; //4 antecedents
                                //if
rule[7].ant_input_index[0]:=0; //AREA
                                //is
rule[7].ant_mf_index[0]:=1; //SMALL
                                //and
rule[7].ant_input_index[1]:=1; //HOLE
                                //is
rule[7].ant_mf_index[1]:=0; //SMALL
                                //and
rule[7].ant_input_index[2]:=3; //SHAPE
                                //is
rule[7].ant_mf_index[2]:=1; //SMOOTH
                                //and
rule[7].ant_input_index[3]:=2; //PERIMETER
                                //is
rule[7].ant_mf_index[3]:=0; //SHORT
```

```
                                        //then OUTPUT is
rule[7].con_mf_index:=1; //BONE

rule_strength[7]:=1;
//*****************************************************************
//if AREA is MEDIUM and HOLE is SMALL and SHAPE is SMOOTH and
//      PERIMETER is SHORT and ROUNDNESS is MEDIUM then OUTPUT is BONE
no_rules_ant[8]:=4; //5 antecedents
                                          //if
rule[8].ant_input_index[0]:=0; //AREA
                                          //is
rule[8].ant_mf_index[0]:=2; //MEDIUM
                                          //and
rule[8].ant_input_index[1]:=1; //HOLE
                                          //is
rule[8].ant_mf_index[1]:=0; //SMALL
                                          //and
rule[8].ant_input_index[2]:=3; //SHAPE
                                          //is
rule[8].ant_mf_index[2]:=1; //SMOOTH
                                          //and
rule[8].ant_input_index[3]:=2; //PERIMETER
                                          //is
rule[8].ant_mf_index[3]:=0; //SHORT
                                          //and
rule[8].ant_input_index[4]:=4; //ROUNDNESS
                                          //is
rule[8].ant_mf_index[4]:=1; //MEDIUM
                                          //then OUTPUT is
rule[8].con_mf_index:=1; //BONE

rule_strength[8]:=1;
//*****************************************************************
//if AREA is BIG and HOLE is SMALL and SHAPE is SMOOTH and
//      PERIMETER is SHORT and ROUNDNESS is MEDIUM then OUTPUT is BONE
no_rules_ant[9]:=4; //5 antecedents
                                          //if
rule[9].ant_input_index[0]:=0; //AREA
                                          //is
rule[9].ant_mf_index[0]:=3; //BIG
                                          //and
rule[9].ant_input_index[1]:=1; //HOLE
                                          //is
rule[9].ant_mf_index[1]:=0; //SMALL
                                          //and
rule[9].ant_input_index[2]:=3; //SHAPE
                                          //is
rule[9].ant_mf_index[2]:=1; //SMOOTH
                                          //and
rule[9].ant_input_index[3]:=2; //PERIMETER
                                          //is
rule[9].ant_mf_index[3]:=0; //SHORT
                                          //and
rule[9].ant_input_index[4]:=4; //ROUNDNESS
                                          //is
rule[9].ant_mf_index[4]:=1; //MEDIUM
                                          //then OUTPUT is
rule[9].con_mf_index:=1; //BONE

rule_strength[9]:=1;
//*****************************************************************
//if ROUNDNESS is LOW then OUTPUT is NOBONE
no_rules_ant[10]:=0; //1 antecedents
                                          //if
```

```
rule[10].ant_input_index[0]:=4; //ROUNDNESS
                                  //is
rule[10].ant_mf_index[0]:=0; //LOW
                               //then OUTPUT is
rule[10].con_mf_index:=0; //NOBONE

rule_strength[10]:=1;


if form1.small.checked then
                          begin
                               rule_strength[7]:=1;
                               rule_strength[8]:=1;
                               rule_strength[9]:=1;
                          end;
if form1.medium.checked then
                          begin
                               rule_strength[7]:=0;
                               rule_strength[8]:=1;
                               rule_strength[9]:=1;
                          end;
if form1.large.checked then
                          begin
                               rule_strength[7]:=0;
                               rule_strength[8]:=0;
                               rule_strength[9]:=1;
                          end;

end; //end of FUzzy_Initialise;

//*************************************************************************8
//**************************************************************************
//***************************************************************************
//**************************************************************************
//**************************************************************************
procedure Fuzzy_Initialise2;
var sir:string;
    code:integer;
begin

no_inputs:=1;   //2 inputs BRIGHTNESS, DIFFERENCE in this order

//the values on the top will be updated every time we make a FuzzyDecision with
//the measured values of the current area in question

no_input_mf[0]:=1; //2 mf for BRIGHTNESS
no_input_mf[1]:=2; //3 mf for DIFFERENCE

//*************************************************************************
input_mf[0,0].a:=0;   //BRIGHTNESS LOW
input_mf[0,0].b:=0;
input_mf[0,0].c:=175;
input_mf[0,0].d:=175;

input_mf[0,1].a:=165;    //BRIGHTNESS HIGH
input_mf[0,1].b:=165;
input_mf[0,1].c:=255;
input_mf[0,1].d:=255;


//*************************************************************************
input_mf[1,0].a:=0; //DIFFERENCE BIG
input_mf[1,0].b:=0;
input_mf[1,0].c:=237;
input_mf[1,0].d:=240;
```

```
input_mf[1,1].a:=238; //DIFFERENCE SMALL
input_mf[1,1].b:=240;
input_mf[1,1].c:=254;
input_mf[1,1].d:=255;

input_mf[1,2].a:=254;   //DIFFERENCE NEGATIVE
input_mf[1,2].b:=255;
input_mf[1,2].c:=510;
input_mf[1,2].d:=510;
//*********************************************************************

//define the output mf
no_output_mf:=1;  //2 mf for output

output_mf[0].a:=0;    //NO_BONE
output_mf[0].b:=0;
output_mf[0].c:=0;
output_mf[0].d:=75;

output_mf[1].a:=25;    //BONE
output_mf[1].b:=100;
output_mf[1].c:=100;
output_mf[1].d:=100;



//*****************************************************************
//if DIFFERENCE is SMALL and BRIGHTNESS is LOW then OUTPUT is NOBONE
no_rules_ant[0]:=1;  //2 antecedents
                                        //if
rule[0].ant_input_index[0]:=1;  //DIFFERENCE
                                        //is
rule[0].ant_mf_index[0]:=1;  //SMALL
                                        //and
rule[0].ant_input_index[1]:=0;  //BRIGHTNESS
                                        //is
rule[0].ant_mf_index[1]:=0;  //LOW
                                        //then OUTPUT is
rule[0].con_mf_index:=0;  //NOBONE

rule_strength[0]:=1;
//*****************************************************************
//if DIFFERENCE is NEGATIVE then OUTPUT is NO_BONE
no_rules_ant[1]:=0;  //1 antecedents
                                        //if
rule[1].ant_input_index[0]:=1;  //DIFFERENCE
                                        //is
rule[1].ant_mf_index[0]:=2;  //NEGATIVE
                                        //then OUTPUT is
rule[1].con_mf_index:=0;  //NOBONE

rule_strength[1]:=1;
//*****************************************************************
//if DIFFERENCE is BIG then OUTPUT is BONE
no_rules_ant[2]:=0;  //1 antecedents
                                        //if
rule[2].ant_input_index[0]:=1;  //DIFFERENCE
                                        //is
rule[2].ant_mf_index[0]:=0;  //BIG
                                        //then OUTPUT is
rule[2].con_mf_index:=1;//BONE

rule_strength[2]:=1;
//*****************************************************************
//if DIFFERENCE is SMALL and BRIGHTNESS is HIGH then OUTPUT is BONE
```

```
no_rules_ant[3]:=1; //2 antecedents
                                        //if
rule[3].ant_input_index[0]:=1; //DIFFERENCE
                                        //is
rule[3].ant_mf_index[0]:=1; //SMALL
                                        //and
rule[3].ant_input_index[1]:=0; //BRIGHTNESS
                                        //is
rule[3].ant_mf_index[1]:=1; //HIGH
                                        //then OUTPUT is
rule[3].con_mf_index:=1; //BONE

rule_strength[3]:=0;


if form1.all.checked then rule_strength[3]:=1;
end; //end of FUzzy_Initialise2;


//fuzzy module 1 for the geometrical filtering
procedure FuzzyDecision(area,hole:integer);
var i,j,l:integer;
    sir:string;
begin
Fuzzy_Initialise;
no_inputs:=4; //we have 5 fuzzy inputs , area and hole
crisp_input[0]:=area;
crisp_input[1]:=hole;
crisp_input[2]:=perimeter1;
crisp_input[3]:=shape;
crisp_input[4]:=roundness;


for i:=0 to no_inputs do
     FuzzifyInput(i); //fuzzify the input i
 for i:=0 to no_rules do
     EvaluateRule(i);
 DefuzzifyOutput;

end;


//fuzzy module 2 for the statistical filtering
procedure FuzzyDecision2;
var i,j,l:integer;
    sir:string;
begin
Fuzzy_Initialise2;
no_inputs:=2; //we have 3 fuzzy inputs
crisp_input[0]:=brightness;
crisp_input[1]:=difference;



 for i:=0 to no_inputs do
     FuzzifyInput(i); //fuzzify the input i
 for i:=0 to no_rules do
     EvaluateRule(i);
 DefuzzifyOutput;

end;

// procedure for colouring the background
procedure background_variant;
var i,j:integer;
begin
```

```
//first scan de image line by line downwards until finds and edge
for i:=0 to n do
begin
      for j:=0 to n do
      begin
      if (copy[i,j]=0) or (copy[i,j]=8) then copy[i,j]:=8;
      if (copy[i,j]=1) then break; // we find the edge
      end;
end;
// now scan upwards until edge
for i:=0 to n do
begin
      for j:=n downto 0 do
      begin
      if (copy[i,j]=0) or (copy[i,j]=8) then copy[i,j]:=8;
      if (copy[i,j]=1) then break; // we find the edge
      end;
end;

// now scan from left to right
for j:=0 to n do
begin
      for i:=0 to n do
      begin
      if (copy[i,j]=0) or (copy[i,j]=8) then copy[i,j]:=8;
      if (copy[i,j]=1) then break; // we find the edge
      end;
end;

//now scan from right to left
for j:=0 to n do
begin
      for i:=n downto 0 do
      begin
      if (copy[i,j]=0) or (copy[i,j]=8) then copy[i,j]:=8;
      if (copy[i,j]=1) then break; // we find the edge
      end;
end;


end;        // end of background_variant


//*************************************************************************
//*************************************************************************
//*************************************************************************
//checks if the contour contains air(holes)
function Check_Air:boolean;
var i,j,air_size:integer;
    holes,airflag:boolean;
    begin
Check_Air:=false;
airflag:=false;
//cover the background first of the mask
for i:=0 to n do
for j:=0 to n do
    copy[i,j]:=mask[i,j]; // save the mask and work with its copy

    background_variant; // colour the background; start from (5,5) because there is white for sure
//now we end up with the copy matrix containing the contour(segment=1) and the
// background (=8).; if there are zeros in copy, it means air is trapped inside
// and we discard the contour as not being valid

//there still can be air trapped into something concave
//find the first white with one neighbour that is 8
for i:=0 to n do
```

```
for j:=0 to n do
    begin
    if (copy[i,j]=0) then
                        begin
                            if (copy[i-1,j-1]=8) or
                               (copy[i-1,j]=8) or
                               (copy[i-1,j+1]=8)or
                               (copy[i,j-1]=8)or
                               (copy[i,j+1]=8)or
                               (copy[i+1,j-1]=8)or
                               (copy[i+1,j]=8) or
                               (copy[i+1,j+1]=8) then background(i,j);
                        end;
   end;


for i:=0 to n do
for j:=0 to n do
    air[i,j]:=copy[i,j]; // work with the air matrix

//find the air if exists
if Form1.CheckBox1.Checked then
begin
repeat
airflag:=false;
for i:=0 to n do
begin
for j:=0 to n do
    if air[i,j]=0 then begin
                         airflag:=true;
                         break;
                         end;
    if airflag then break;
end;

if airflag then //there is air inside the current contour
begin
  //check if the air size for the holes is larger than 30
  // if it's not make the air bone and concatenate regions
  //if it's larger than 30-40ish than IT IS PROPER AIR
  //now we work with the air matrix in which background is 8, contour is 1 and air 0

      Air_hole(i,j); //deal with the currrent airhole
   //compute the size of the current airhole
     air_size:=0;
     for i:=0 to n do
     for j:=0 to n do
             if (air[i,j]=7) then air_size:=air_size+1;
     if (air_size<AIRHOLE_SIZE) then
             //if it is an accepted size of the airhole then merge it with the contour
        begin
            Check_Air:=false;
            for i:=0 to n do
            for j:=0 to n do
                if (air[i,j]=7) then begin air[i,j]:=1; //merge the hole with the contour
                                           final[i,j]:=1;
                                           end;


        end
     else  begin
                 Check_Air:=true;
                 break;
                 end;
end; //end of the air if

until not airflag;   //if there is air but can be merged we need to find the next air pouch , so
```

```
                    //repeat until there is no more air or there is at least one that
                    //is considered PROPER air and there is no point in continue to search
                    //for the next air pouch
end
else //no neeed to merge the air holes
begin
     Check_air:=false;
    for i:=0 to n do
    for j:=0 to n do
        if air[i,j]=0 then Check_Air:=true;

end;

end;


//*************************************************************************
//*************************************************************************
//*************************************************************************
//measure the charactiristics of the current object
procedure Measure_contour;
var i,j,area:integer;
    sir:string;
begin
     area:=0;
     for i:=0 to n do
     for j:=0 to n do
             if (mask[i,j]=1) then area:=area+1;
     if (area<MAX_BONE_SIZE) and (area>=MIN_BONE_SIZE) then
             //possible bone - save the current contour/segment into final
        begin
        for i:=0 to n do
        for j:=0 to n do
            if (mask[i,j]<>0) then final[i,j]:=mask[i,j];
        end
        else DrawSegmentRed;

//finished with the current contour
//erase it from the input matrix

for i:=0 to n do
for j:=0 to n do
    begin
        if (mask[i,j]<>0) then input[i,j]:=0;
    end;


end; //end of Measure_contour



//procedure to backtracking the perimeter
//in the mask matrix put 100 where a pixels belongs to the perimeter
procedure Perimeter(x,y:integer);
begin
if (mask[x,y]<>0) and (mask[x,y]<>100)
   and ((mask[x+1,y]=0) or (mask[x+1,y+1]=0) or (mask[x,y+1]=0) or (mask[x-
1,y]=0) or (mask[x-1,y-1]=0) or (mask[x,y-1]=0))
                    then //pixel in the contour
                  begin
                      mask[x,y]:=100;
            if (mask[x+1,y]<>0) and (mask[x+1,y]<>100) then perimeter(x+1,y);
            if (mask[x-1,y]<>0) and (mask[x-1,y]<>100) then perimeter(x-1,y);
            if (mask[x,y+1]<>0) and (mask[x,y+1]<>100) then perimeter(x,y+1);
            if (mask[x,y-1]<>0) and (mask[x,y-1]<>100) then perimeter(x,y-1);
```

```
            if    (mask[x+1,y+1]<>0)    and    (mask[x+1,y+1]<>100)    then
            perimeter(x+1,y+1);
            if (mask[x-1,y-1]<>0) and (mask[x-1,y-1]<>100) then perimeter(x-
            1,y-1);

                  end;
end;


procedure CalculatePerimeter;
var i,j:integer;
    x,y:integer;
begin
      //find the first point of the area
    for i:=0 to n do
    begin
    for j:=0 to n do
        if mask[i,j]<>0 then
           begin
                x:=i;
                y:=j;
                break;
           end;
        if mask[i,j]<>0 then break;
    end;
    Perimeter(x,y);
end;


//function to determine wheather a pixel (x,y) belongs to a circle
//with the center in (xcenter,ycenter) of radius ref_radius
function InCircle(var xcenter,ycenter,x,y:integer;radius:real):boolean;
begin
 if (sqr(x-xcenter)+sqr(y-ycenter)>sqr(radius)) then InCircle:=false
 else InCircle:=true;
end;



//function to extract geometrical features from the current object
procedure MeasureGeometricalCharacteristics(area,xmin,ymin,xmax,ymax:integer);
var i,j:integer;
    ref_radius:real; //the theoretical reference radius
    xcenter,ycenter:integer; //center for the theoretical circle with the same size as the region
    not_ref_area,difference_area:integer;
    not_area:integer;


begin
    brightness1:=0;
    brightness2:=0;
    contrast1:=0;
    contrast2:=0;
    for i:=0 to n do
    for j:=0 to n do
        if mask[i,j]<>0 then
                          begin
                              brightness1:=brightness1+in1[i,j]/area;
                              brightness2:=brightness2+in2[i,j]/area;
                          end;
    for i:=ymin to ymax do
    for j:=xmin to xmax do
            begin
                if mask[i,j]=0 then
                begin
            contrast1:=contrast1+in1[i,j]/((ymax-ymin+1)*(xmax-xmin+1)-area);
            contrast2:=contrast2+in2[i,j]/((ymax-ymin+1)*(xmax-xmin+1)-area);
```

```
                end;
            end;
CalculatePerimeter;
perimeter1:=0;
for i:=0 to n do
for j:=0 to n do
    if mask[i,j]=100 then perimeter1:=perimeter1+1;

//now compute the reference center for the reference circle
xcenter:=((xmax-xmin) div 2)+xmin;
ycenter:=((ymax-ymin) div 2)+ymin;
//compute the reference radius for the area in question area=2PI R
ref_radius:=sqrt(area/PI);

difference_area:=0;
//the equation of the reference circle is (x-xcenter)^2+(y-ycenter)^2=radius^2
//compute the difference between the two areas (are ain question and the reference one)
for i:=0 to n do
for j:=0 to n do
    begin
        if (mask[i,j]<>0) and (not InCircle(xcenter,ycenter,j,i,ref_radius))
then
            //it means this pixel belongs to the area in question and is not in the reference circle
                begin
                    difference_area:=difference_area+1;
                end;
        if (mask[i,j]=0) and InCircle(xcenter,ycenter,j,i,ref_radius) then
        //this pixel belongs to the reference circle but not to the area in euqstion
                begin
                   difference_area:=difference_area+1;
                end;
    end;

//now we can compute the roundness measure for the current area
roundness:=(1-difference_area/(2*area))*100;
//compute the shape measures
shape:=(area/perimeter1)*10;
if shape>100 then shape:=99;
end;



//procedure for measuring statistical grey-level characteristics of the current object
procedure MeasureStatisticalCharacteristics(area,xmin,ymin,xmax,ymax:integer);
var i,j:integer;
    std_dev1,std_dev2:real;
begin
    brightness1:=0;
    brightness2:=0;
    contrast1:=0;
    contrast2:=0;
    for i:=ymin to ymax do
    for j:=xmin to xmax do
        if mask[i,j]<>0 then
                            begin
                                brightness1:=brightness1+in1[i,j];
                                brightness2:=brightness2+in2[i,j];
                            end;
    brightness1:=brightness1/area;
    brightness2:=brightness2/area;
    if brightness1>brightness2 then brightness:=brightness1
                               else brightness:=brightness2;
 //we now have the brightness; let's compute the standard deviation as the measure
 //for togetherness
    std_dev1:=0;
    std_dev2:=0;
    for i:=ymin to ymax do
```

```
    for j:=xmin to xmax do
        if mask[i,j]<>0 then
                        begin
                             std_dev1:=std_dev1+(sqr(in1[i,j]-brightness1));
                             std_dev2:=std_dev2+(sqr(in2[i,j]-brightness2));
                        end;
    togetherness1:=sqrt(std_dev1/area);
    togetherness2:=sqrt(std_dev2/area);

  //compute the mean local contrast without the area
    for i:=ymin to ymax do
    for j:=xmin to xmax do
            begin
                if mask[i,j]=0 then
                begin
                     contrast1:=contrast1+in1[i,j];
                     contrast2:=contrast2+in2[i,j];
                end;
            end;
    contrast1:=contrast1/((ymax-ymin+1)*(xmax-xmin+1)-area);
    contrast2:=contrast2/((ymax-ymin+1)*(xmax-xmin+1)-area);
    if contrast1>contrast2 then contrast:=contrast2
                           else contrast:=contrast1;
    difference:=255-(brightness1-contrast1+brightness2-contrast2)/2
end;



//*************************************************************************
//*************************************************************************
//*************************************************************************
//FUZZY measure the charactiristics of the data
procedure Fuzzy_Measure_contour;
var i,j,area:integer;
    sir:string;
    air_size:integer;
    holes,airflag:boolean;
    possible:boolean;
    xmin,xmax,ymin,ymax:integer;
begin
//determine the size of the current area
     area:=0;
     for i:=0 to n do
     for j:=0 to n do
             if (mask[i,j]=1) then area:=area+1;

//determine airhole sizes and apply the fuzzydecisions

//determine AIR HOLES and their sizes
airflag:=false;
//cover the background first of the mask
for i:=0 to n do
for j:=0 to n do
    copy[i,j]:=mask[i,j]; // save the mask and work with its copy

    background_variant; // colour the background; start from (5,5) because there is white for sure
//now we end up with the copy matrix containing the contour(segment=1) and the
// background (=8).; if there are zeros in copy, it means air is trapped inside
// and we discard the contour as not being valid

//there still can be air trapped into something concave
//find the first white with one neighbour that is 8
for i:=0 to n do
for j:=0 to n do
    begin
    if (copy[i,j]=0) then
```

```
                        begin
                            if (copy[i-1,j-1]=8) or
                               (copy[i-1,j]=8) or
                               (copy[i-1,j+1]=8)or
                               (copy[i,j-1]=8)or
                               (copy[i,j+1]=8)or
                               (copy[i+1,j-1]=8)or
                               (copy[i+1,j]=8) or
                               (copy[i+1,j+1]=8) then background(i,j);
                        end;
    end;


for i:=0 to n do
for j:=0 to n do
    air[i,j]:=copy[i,j]; // work with the air matrix

possible:=false; //start with the assumption that is not a BONE

//find if there is air
for i:=0 to n do
begin
for j:=0 to n do
    if air[i,j]=0 then begin
                        airflag:=true;
                        break;
                        end;
    if airflag then break;
end;

//if there is air, find it and repeat until no more air is found
if airflag then
begin
repeat //for all the airholes found
airflag:=false;
air_size:=0;
for i:=0 to n do
begin
for j:=0 to n do
    if air[i,j]=0 then begin
                        airflag:=true;
                        break;
                        end;
    if airflag then break;
end;

if airflag then //there is air inside the current contour
begin
    //check if the air size for the holes is larger than 30
  // if it's not make the air bone and concatenate regions
  //if it's larger than 30-40ish than IT IS PROPER AIR
  //now we work with the air matrix in which background is 8, contour is 1 and air 0

      Air_hole(i,j); //deal with the currrent airhole
   //compute the size of the current airhole
     air_size:=0;
     for i:=0 to n do
     for j:=0 to n do
             if (air[i,j]=7) then air_size:=air_size+1;

        //compute other characteristics of the area
    //first find xmin,xmax,ymin,ymax to draw a rectangle for the current area
        xmin:=536;xmax:=0;ymin:=536;ymax:=0;
        for i:=0 to n do
        for j:=0 to n do
            begin
```

```
                if (mask[i,j]<>0) then
                   begin
                       if xmin>j then xmin:=j;
                       if xmax<j then xmax:=j;
                       if ymin>i then ymin:=i;
                       if ymax<i then ymax:=i;
                   end;
            end;
    //measure characteristics roundness, perimeter
         MeasureGeometricalCharacteristics(area,xmin,ymin,xmax,ymax);
    //apply the fuzzy module 1 for the geometrical filtering
     FuzzyDecision(area,air_size);
     if crisp_output>65 then //if it is an accepted size of the airhole then merge it with the contour
        begin
            possible:=true;
            for i:=0 to n do
            for j:=0 to n do
                if (air[i,j]=7) then begin air[i,j]:=1; //merge the hole with the contour
                                           final[i,j]:=1;
                                           mask[i,j]:=7;
                                           end;

        end
     else  begin
                possible:=false;
                break;
                end;
end; //end of the air if

if not possible then break; //exit the loop if at least one is NOBONE

until not airflag;      //if there is air but can be merged we need to find the next air pouch , so
                        //repeat until there is no more air or there is at least one that
          //is considered PROPER air and there is no point in continue to search
          //for the next air pouch
end //end of the air is found at least once
else //if there is no air in the segment
     begin
        //compute other characteristics of the area
    //first find xmin,xmax,ymin,ymax to draw a rectangle for the current area
        xmin:=536;xmax:=0;ymin:=536;ymax:=0;
        for i:=0 to n do
        for j:=0 to n do
            begin
                if (mask[i,j]<>0) then
                   begin
                       if xmin>j then xmin:=j;
                       if xmax<j then xmax:=j;
                       if ymin>i then ymin:=i;
                       if ymax<i then ymax:=i;
                   end;
            end;
    //measure characteristics roundness, perimeter
         MeasureGeometricalCharacteristics(area,xmin,ymin,xmax,ymax);
    //apply the fuzzy module 1 for the geometrical filtering
         FuzzyDecision(area,0);
         if crisp_output>65 then possible:=true
                            else possible:=false;
     end;

if possible then //possible bone - save the current contour/segment into final
   begin

    //apply the second fuzzy module
        MeasureStatisticalCharacteristics(area,xmin,ymin,xmax,ymax);
        FuzzyDecision2;
```

```
        if crisp_output>65 then //if there is indeed a bone save it into the final matrix and image
        begin
        no_bones:=no_bones+1;
    //save the current area into the final matrix
        for i:=0 to n do
        for j:=0 to n do
            if (mask[i,j]<>0) then final[i,j]:=mask[i,j];
        end
    end

//finished with the current contour
//erase it from the input matrix
for i:=0 to n do
for j:=0 to n do
    begin
        if (mask[i,j]<>0) then input[i,j]:=0;
    end;


end; //end of Fuzzy_Measure_contour


function Check_More_Contours:boolean; //check if there are still contours to be processed
var i,j:integer;
begin
Check_More_Contours:=false;
for i:=0 to n do
begin
for j:=0 to n do
    if (input[i,j]<>0) then
                        begin
                        Check_more_Contours:=true;
                        break;
                        end;
    if Check_More_Contours then break;
end;
end; // end of Check_more_contours
```

C.3 Screenshots

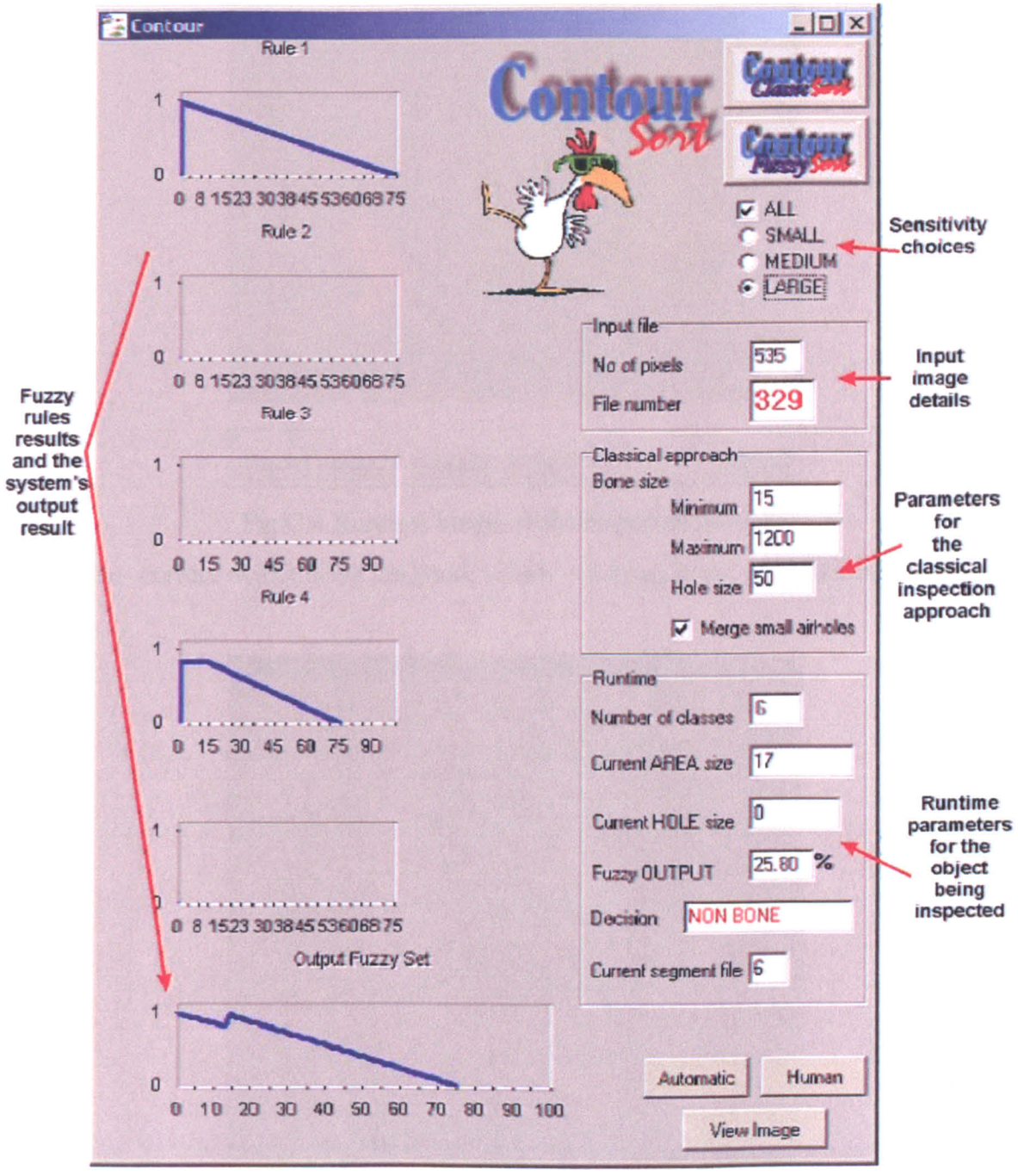


Fig. C.3 Main window of the application

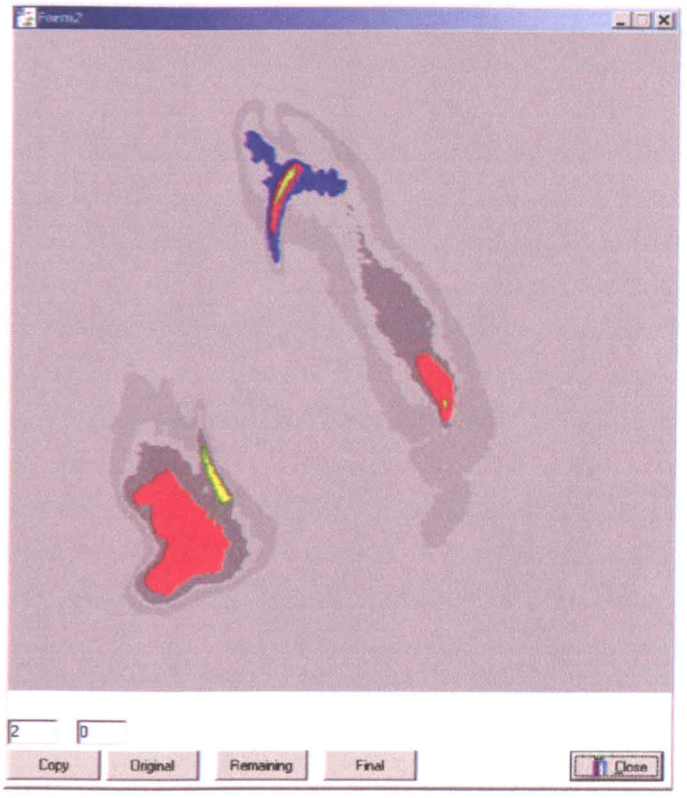


Fig.C.4 Runtime image of the inspected product

(blue- current object being analysed; yellow – foreign body; red – normal meat)

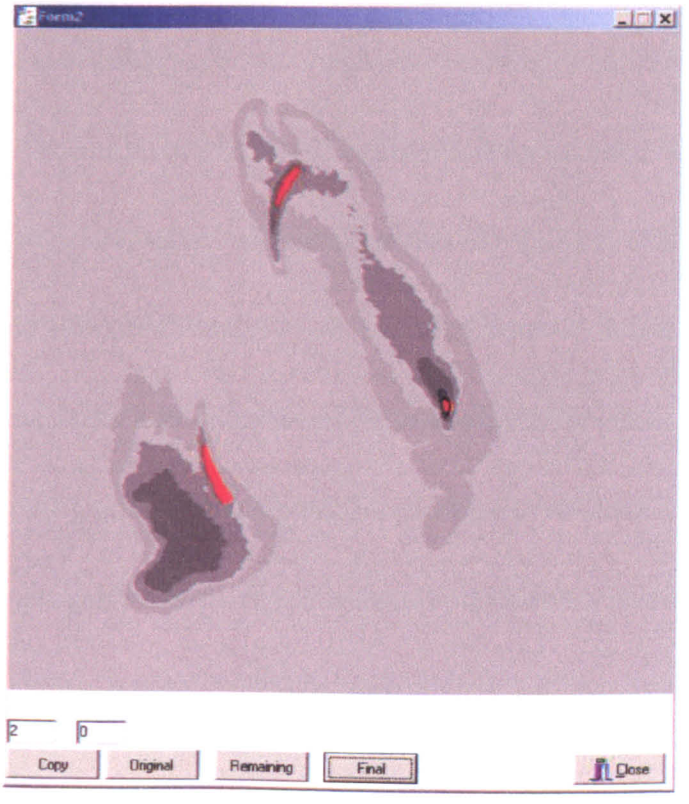


Fig.C.5 Final result of the inspection

(red – foreign bodies detected)